

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE
—

TESI DI LAUREA TRIENNALE IN INGEGNERIA BIOMEDICA

GENERAZIONE DEL FEEDBACK
SONORO NELL'ESECUZIONE DI
TASK MOTORI

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: ANDREA GAZANEO

ANNO ACCADEMICO 2009-2010

ai miei genitori e a tutti i miei amici che mi hanno sostenuto in questi anni..

Indice

Sommario	VII
Introduzione	IX
1 Feedback sonoro nella neuroriabilitazione	1
1.1 Obiettivi	3
1.2 Prima parte: Software di elaborazione dei dati	3
1.3 Seconda parte: Test su soggetti sani	4
2 Sintesi del suono	7
2.1 Protocollo OSC	7
2.1.1 Sintassi	8
2.1.2 OSC Packets	8
2.1.3 OSC Semantics	10
2.2 Il nostro caso	12
2.3 Pure Data	14
2.3.1 Pure Data applicato alla mia interfaccia	16
3 Software sviluppato	19
3.1 Implementazione interfaccia	20
3.2 Istruzioni preliminari per l'utilizzo dell'interfaccia	21
3.2.1 Istruzioni: "ESECUZIONE TEST - ONLINE"	22
3.2.2 Visualizzazione ed invio dei dati a Pure Data	34
3.2.3 Istruzioni: "RIPRODUZIONE TEST - OFFLINE"	36

4	Test sperimentali	39
4.1	Pianificazione	39
4.2	Esecuzione Task	44
5	Elaborazione dati test	47
5.1	Estrazioni dei dati significativi con Matlab	47
5.2	Elaborazione dei dati con software statistico	52
5.2.1	GraphPad Prism	52
5.2.2	Test di Wilcoxon	52
5.2.3	Il nostro caso	53
	Conclusioni	59
	Bibliografia	61

Sommario

Il mio progetto di tesi si può suddividere essenzialmente in due parti.

Nella parte iniziale mi sono dedicato all'implementazione di un'interfaccia GUI, tramite il software Matlab, per l'esecuzione di task motori basati sul movimento dell'esoscheletro Pneu-WREX e il contemporaneo invio di dati, via UDP, al software Pure-Data, una piattaforma per la sintesi e l'elaborazione audio in tempo reale.

Nella seconda parte invece ho testato l'interfaccia GUI da me creata su 20 soggetti sani per verificare se il feedback audio è in grado di agevolare le loro prestazioni nell'esecuzione del test.

Introduzione

Contesto: epidemiologia ictus

Secondo la definizione dell'OMS (Organizzazione Mondiale della Sanità) l'ictus è l'improvvisa comparsa di segni e/o sintomi riferibili a deficit focale e globale delle funzioni cerebrali, di durata superiore alle 24 ore. Questa patologia è una emergenza medica e deve essere prontamente diagnosticata e trattata in un ospedale per l'elevato rischio di disabilità e di morte che esso comporta. La definizione di ictus comprende, sulla base dei dati morfologici, l'ictus ischemico più frequente, l'ictus emorragico nel 15% dei casi, e alcuni casi di emorragia meningea.

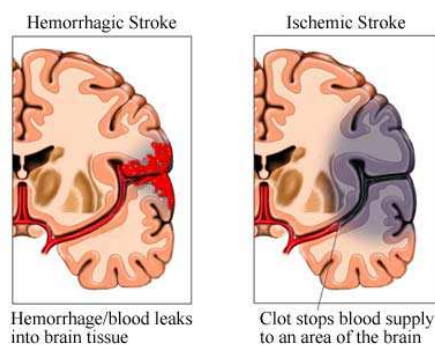


Figura 1: Rappresentazione grafica di un ictus emorragico e di un ictus ischemico

Nei Paesi industrializzati l'ictus rappresenta, per le sue dimensioni epidemiologiche e anche per il suo impatto socio-economico, una delle più importanti problematiche sanitarie, costituendo la prima causa di invalidità permanente e la seconda causa di demenza. I dati a riguardo sono a dir poco allarmanti: ogni anno si hanno 4.5 milioni di morti nel mondo di cui 400.000 solo in Europa a causa di questa grave patologia.

Nel nostro paese le malattie cerebrovascolari in generale hanno causato 246.415 ricoveri pari al 36.7% del totale dei ricoveri per malattie del sistema nervoso centrale e al 2.7% di tutti i ricoveri. Questa patologia può verificarsi ad ogni età, ma nel 50% dei casi i soggetti con ictus cerebrale hanno più di 70 anni, con una percentuale leggermente più alta negli uomini (7.4%) rispetto alle donne (5.9%)[1]. Il rischio di esserne colpiti aumenta con l'età: in pratica raddoppia ogni 10 anni a partire dai 45 anni, raggiungendo il valore massimo negli ultra-80enni [2]. Il progressivo invecchiamento della popolazione e la considerazione che l'incidenza dell'ictus è strettamente collegata all'età, fanno ritenere che in Italia l'incidenza di eventi ictali aumenterà nei prossimi anni: le proiezioni fino al 2016 fanno ritenere che tale aumento sarà circa del 22.2%.

L'aumento del tempo di sopravvivenza dopo l'ictus porterà a un aumento della prevalenza di tale patologia e della domanda di servizi sanitari per pazienti cronici stabilizzati con handicap neurologici. Il 40-50% dei soggetti che sopravvivono perde definitivamente la propria autonomia motoria mentre il 10% ha una recidività grave entro 12 mesi. In Italia oltre la metà dei sopravvissuti ha una disabilità nelle ADLs (Activities of Daily Living) . Secondo i dati di incidenza italiani si possono stimare circa 130.000 nuovi casi all'anno di ictus, circa la metà dei quali con disabilità a 6 mesi [3][4]. E' opinione comune che i risultati della riabilitazione riguardo al danno motorio e funzionale sono strettamente legati al tempo che intercorre tra l'evento traumatico (l'ictus) e l'inizio della terapia.

Di fondamentale importanza risulta quindi un intervento specifico e immediato nella fase acuta, immediatamente successiva all'ictus. In passato l'assistenza al soggetto con ictus era affrontata con atteggiamento passivo.

Negli ultimi anni invece, dopo la pubblicazione e l'esperienza degli studi clinici che mostravano esiti migliori nei soggetti trattati in centri specifici, si sono diffuse diverse unità organizzate per affrontare questa condizione, le cosiddette Stroke Unit. Pur nella variabilità degli assetti organizzativi e assistenziali sperimentali, le analisi effettuate hanno suggerito che possono essere considerati elementi indispensabili della Stroke Unit:

- La multidisciplinarietà di approccio da parte di un team esperto in malattie cerebrovascolari;

- La tempestività di avvio dell'intervento riabilitativo;
- L'applicazione di protocolli espliciti per la prevenzione delle complicanze maggiori;
- La messa in atto di interventi mirati alla formazione e all'aggiornamento continuo del personale;
- Il coinvolgimento di chi assiste il soggetto nel percorso riabilitativo;

Bisogna sottolineare però che in Italia queste strutture specializzate sono ancora molto poche. Se ne contano 89 in tutto il territorio nazionale di cui solo 8 nell'Italia meridionale. La realtà è che i pazienti sono sottoposti solo ad una terapia hands-on, ovvero a diretto contatto con il fisioterapista. Purtroppo la qualità e la quantità delle terapie a cui si sottopone un soggetto si dimostra essere limitata dai costi che queste comportano e dalla disponibilità dei fisioterapisti nelle varie strutture ospedaliere. Risulta necessario, quindi, introdurre delle nuove terapie a basso costo che permettano al paziente di svolgere le sue attività riabilitative per periodi più lunghi e in modo continuato soprattutto nel periodo di fase acuta di questa patologia.

In questo background si inseriscono le numerose sperimentazioni dei due ultimi decenni. La ricerca in questi ultimi anni ha portato a creare e ottimizzare delle nuove terapie riabilitative basate sull'uso di sistemi robotici. Queste tecniche innovative affiancate alla terapia tradizionale seguita da un fisioterapista possono migliorare le performance motorie del paziente e accorciarne i tempi di recupero. Un altro importante beneficio della riabilitazione tramite dispositivi robotici risiede nel fatto che si può quantificare, attraverso parametri analitici, se un paziente sta effettivamente rispondendo alla terapia a cui è sottoposto.

Dispositivi robotici per riabilitazione

Nell'ambito della riabilitazione dei pazienti affetti da ictus, lo sviluppo della ricerca si è indirizzato verso alcuni strumenti tecnologici innovativi. Tra le varie tecnologie la sperimentazione di alcuni sistemi robotizzati per il recupero della funzionalità motoria dell'arto superiore è di particolare interesse sia per la iniziale

dimostrazione di efficacia clinica, evidenziata da precedenti pubblicazioni [5], sia per i futuri esiti che potrebbe avere in seguito allo sviluppo di nuove tecnologie.

I robot adibiti alla riabilitazione motoria si possono suddividere in due famiglie principali:

1. Esoscheletri
2. End effector base

I robot di prima generazione sono stati gli esoscheletri che però sono risultati assolutamente non modulabili, per le esigenze cliniche dei vari pazienti, molto costosi e particolarmente ingombranti e pesanti, per cui sono sostanzialmente stati abbandonati. Alcuni esempi di esoscheletri sono: il Pneu-WREX, in grado di esercitare un ampio range di forze sul braccio quando esso compie movimenti “naturalì” e l’ARMin.



Figura 2: Immagine Pneu-WREX

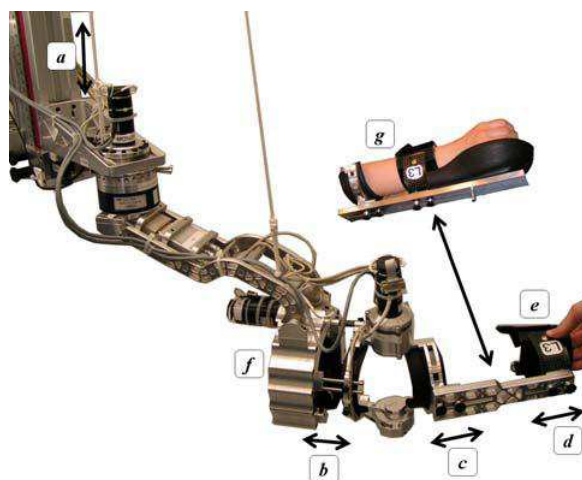


Figura 3: Rappresentazione dell’ARMin e dei diversi punti di regolazione (a, b, c, d, e, f, g)

La seconda generazione di robot ha visto la nascita di alcuni “bracci robotizzati” conosciuti anche come End effector base. Questi dispositivi permettono di interagire in modo del tutto sicuro fornendo al soggetto una sensazione di forza (feedback propriocettivo), una mobilizzazione passiva oppure assistenza alla mobilizzazione, una stimolazione neuro-sensoriale attraverso un sistema di feedback visivo ed uditivo basato sulla realtà virtuale. Tali sistemi sono in grado di misurare le caratteristiche cinematiche ed anche dinamiche del movimento dell’arto superiore con cui interagiscono.

Un esempio molto significativo di questa famiglia di dispositivi è il MIT-Manus .

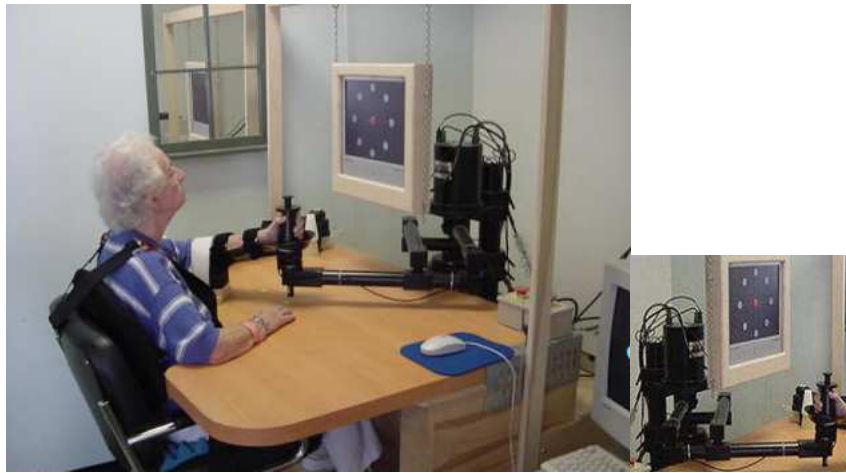


Figura 4: Esecuzione di un tipico esercizio riabilitativo con il MIT Manus

Questa macchina riconosce una componente volontaria del movimento, anche minimale, del paziente e quando questa componente intenzionale viene meno si attiva per aiutare il paziente stesso a terminare il movimento ed a raggiungere il bersaglio assegnato nell’esercizio. Questa particolare soluzione permette di coniugare motricità volontaria residua del malato e raggiungimento del target con un movimento corretto, la combinazione di questi due fattori è fondamentale per riapprendere le sequenze motorie perse dopo l’ictus.

Il progetto di riabilitazione basato su questo dispositivo prevede 3 sedute settimanali di 40 minuti l’una per 6 settimane. I risultati ottenuti sono stati molto significativi dal punto di vista della rieducazione funzionale. Secondo gli esperti, i soggetti coinvolti nel progetto, hanno dimostrato una grandissima motivazione, anche in considerazione del fatto che, in circa 40 minuti di sessione, i pazienti

sono portati a fare un migliaio di movimenti. Altra particolarità tecnica è stata la progettazione dell'apparecchiatura secondo il criterio della “bassissima impedenza”, significa che il paziente non deve trovare nessuna resistenza da parte della macchina e che tutte le opposizioni meccaniche devono essere compensate dai motori per non compromettere la buona riuscita degli esercizi.

Nel settore della robotica riabilitativa anche l'Università di Padova ha dato il proprio contributo. Nei laboratori del DIMEG (Dipartimento di Innovazione Meccanica e Gestionale), infatti, è stato realizzato il Neuro-REhabilitation-roBOT (NeReBot), un interfaccia aptica a cavi, visibile in figura, che può essere utilizzata a letto dal paziente con l'obiettivo di riabilitare i soggetti post stroke in fase acuta.



Figura 5: Immagine del NeReBOT sviluppato nei laboratori del DIMEG

NeReBot è stato ideato e realizzato per effettuare diversi movimenti dell'arto superiore (spalla e gomito), simulando un trattamento hand-over-hand. La mobilizzazione passiva del NeReBot (sui 20 soggetti testati) sembra aver ridotto la debolezza e aver favorito il recupero della forza muscolare di pazienti post-stroke con arto superiore plegico/paretico [6].

Tutte queste nuove tecnologie di tipo meccanico, però, devono essere integrate alla motivazione e al coinvolgimento del paziente nella riabilitazione robot-assistita. Per ottenere questi effetti nel soggetto il metodo migliore è quello di inserire nell'esecuzione dell'esercizio dei feedback audio o visivi. Secondo gli studi

di Reinkensmeyer, un dispositivo robotico integrato con un feedback interattivo può incrementare la motivazione del paziente nell'esecuzione del task motorio [7].

Per quanto riguarda il feedback audio ne parlerò molto più approfonditamente nel Capitolo 1, mentre per quanto riguarda il feedback visivo, possiamo dire che esso è identificato come un riscontro sul display di distruttori, parametri cinematici e punteggi ottenuti dal paziente durante la sessione di riabilitazione. Questo tipo di feedback ha un riscontro psicologico nel paziente stesso in quanto aumenta la sua attenzione sull'esercizio e lo invoglia a migliorare le proprie performance.

Inoltre, la possibilità di fornire feedback della cinematica del movimento promuove l'apprendimento in soggetti sani e in pazienti colpiti da ictus.

Uno dei problemi principali di queste nuove metodiche resta la valutazione dell'efficacia dal punto di vista clinico e la quantificazione dei parametri che esse maggiormente influenzano. Date queste premesse, bisognerebbe istituire un adeguato programma che preveda task motori assistiti attraverso un'interfaccia robotica nei pazienti paretici e con il feedback audio-visivo della performance motoria. Questa serie di sperimentazioni dovrebbe essere svolta su un più ampio ed omogeneo campione di pazienti post-ictus per verificarne la vera efficacia clinica.

Gli obiettivi principali che la robotica riabilitativa deve dunque porsi per diventare un adeguato e duraturo mezzo di riabilitazione sono:

1. Realizzazione di esoscheletri e sistemi aptici integrati a basso costo.
2. Creazione di task motori robot-assistiti nella prima fase post-ictus (fase acuta).
3. Valutazione dell'efficienza della terapia robot assistita in termini di riduzione delle ospedalizzazioni durante il progetto riabilitativo.
4. Valutazione del costo della industrializzazione del sistema robotico.

Capitolo 1

Feedback sonoro nella neuroriabilitazione

Alcune recenti ricerche in ambito neurologico hanno dimostrato che la stimolazione sonora è in grado di aumentare la plasticità celebrale, influenzando specifici meccanismi neuronali che possono contribuire al recupero da un danno neurologico, ad esempio un ictus.

In collegamento agli studi fatti da Sarkamo [8], la rielaborazione delle immagini che fotografano l'attività neuronale ha evidenziato che ascoltare musica produce degli indubbi effetti benefici sul cervello [9][?]. La musica, infatti, mette in contatto tra loro diverse parti del cervello, oltre alla corteccia uditiva, tra le quali: l'area frontale, l'area temporale, l'area parietale e l'area subcorticale. La connessione di tutte queste strutture provoca un aumento dell'attenzione, della memoria e delle funzioni motorie.

Suoni familiari possono facilitare e perfezionare particolari risposte motorie che prima erano associate a questi suoni [10]. Alcuni studi analoghi hanno dimostrato che un feedback sonoro ha prodotto netti miglioramenti in alcune ADLs (la distribuzione del peso nel sedersi, la camminata) in pazienti affetti da malattie neurologiche (Parkinson, SM, Ictus).

In seguito a queste ricerche in campo medico, l'ingegneria riabilitativa ha introdotto nei nuovi dispositivi medici un feedback sonoro che può agevolare il paziente sia nell'apprendimento dell'esercizio che a mantenere alta la propria

concentrazione su ciò che sta svolgendo.

L'auditory feedback, propriamente detto, denota un segnale audio, generato automaticamente dal dispositivo, in risposta ad un'azione dell'utente o in risposta ad uno stato interno del sistema. I feedback possono essere suddivisi in due categorie [11][12]:

- Riguardo la conoscenza dei risultati: danno un'indicazione di come è stato svolto un esercizio o se è stato raggiunto l'obiettivo dell'esercizio stesso;
- Riguardo la conoscenza della performance: danno un'indicazione dei movimenti eseguiti dal paziente nel portare a termine un esercizio.

In merito alla definizione sopra citata, i feedback audio più utilizzati sono quelli che forniscono informazioni sugli errori di movimento del paziente, quindi appartengono alla seconda categoria. Essi hanno duplice funzione :

1. Facilitare l'apprendimento del percorso e del movimento previsto dall'esercizio.
2. Incrementare l'attenzione e la motivazione del paziente durante l'esercizio.

Un esempio molto significativo dell'utilizzo di questi feedback integrati in un robot-riabilitativo è dato dal Pneu-WREX. Al paziente, vincolato all'esoscheletro, veniva chiesto di seguire un target su un display e, allo stesso tempo, di identificare un distruttore che compariva sullo schermo stesso. La presenza del distruttore diminuiva notevolmente le prestazioni del paziente, complicando l'inseguimento del target. L'introduzione di un feedback audio, il quale segnalava il superamento di un certo margine di errore, riportava le prestazioni a livelli soddisfacenti [13].

Come ho illustrato in tutti questi esempi, l'importanza di un efficace feedback audio acquista quindi ancora più valore negli esercizi di riabilitazione robot-assistiti. Il paziente in questa tipologia di riabilitazione, infatti, non ha l'ausilio del fisioterapista e il feedback audio è l'unico sistema che incoraggia il paziente stesso a mantenere la propria attenzione sull'esercizio e, quindi, a migliorarsi sempre.

Il lavoro di ricerca pertanto si pone come obiettivo quello di individuare quali siano i parametri più opportuni ed efficaci a cui legare un feedback (acustico o vi-

suale) in modo che esso possa essere veramente d'aiuto al paziente nell'esecuzione dell'esercizio.

1.1 Obiettivi

L'obiettivo principale del progetto è stato quello di implementare due diversi tipi di feedback audio (che dovranno essere integrati al Pneu-WREX) e testarli su un gruppo di soggetti per verificare se i feedback agevolano o meno l'esecuzione degli esercizi riabilitativi.

Prima di dedicarmi all'analisi e alla scelta di un adeguato feedback, ho dovuto realizzare un'interfaccia grafica tramite un software di elaborazione di dati che fosse simile a quella realizzata in California per il Pneu-WREX.

Realizzata l'interfaccia, con l'aiuto del Prof. Giulio Rosati, del Prof. Federico Avanzini e del suo dottorando Simone Spagnol, mi sono dedicato all'analisi dei feedback più adatti per la mia interfaccia.

Si è trattato quindi di trovare i giusti parametri su cui basare la funzione di feedback, come possono essere gli errori di velocità o di posizione tra paziente e target e di capire in che modo deve essere realizzato il feedback al fine di essere il più utile possibile alla riabilitazione del paziente.

1.2 Prima parte: Software di elaborazione dei dati

Il primo obiettivo da raggiungere è stato quello di realizzare un'interfaccia grafica con il programma Matlab che rispecchiasse quella utilizzata per i test del Pneu-WREX.

Il nostro scopo è stato quello di ricreare un esercizio riabilitativo in cui il paziente dovesse inseguire un target su uno schermo, tramite l'utilizzo di una tavoletta grafica. Il target si muove lungo una traiettoria rettilinea piana che può essere, a seconda della modalità scelta, o a lunghezza costante o a lunghezza variabile.

Grazie alla funzione timer della libreria di Matlab sono riuscito a ricreare il movimento del target e ad implementare due diverse tipologie rettilinee di traiettoria: una a lunghezza fissa e una a lunghezza variabile.

Successivamente mi sono dedicato alla realizzazione del puntatore del paziente con il quale avrebbe dovuto seguire il target. Per fare ciò, sempre utilizzando la precedente funzione di Matlab, ho passato le posizioni del puntatore sulla tavoletta grafica (dispositivo utilizzato per i miei test non avendo a disposizione l'esoscheltro Pneu-WREX) e in questo modo ho ricreato il "pallino verde" che identifica i movimenti del paziente.

Terminata la parte prettamente grafica ho inviato via UDP al software Pure Data un pacchetto dati contenente diversi valori tra i quali: la posizione lungo i 2 assi (x e y) del target, del paziente e della velocità. Grazie a questi dati, abbiamo implementato due tipologie diverse di feedback:

1. Legata alla velocità del target: molto simile ad un suono di rotolamento.
2. Legata all'errore di posizione del paziente rispetto al target.

Questi due tipi di feedback possono essere scelti nell'interfaccia in configurazione spazializzata o meno. La spazializzazione di una o più sorgenti audio consiste nella simulazione di un paesaggio sonoro tridimensionale. All'interno del paesaggio, una vera e propria "scena" da ascoltare, le sorgenti assumeranno una posizione virtuale rispetto all'ascoltatore e potranno anche "muoversi" intorno a lui più o meno velocemente.

Per mancanza di tempo purtroppo abbiamo potuto implementare solo questi due tipi di feedback. Per delle future ricerche, però, sarebbe molto interessante provare a realizzare altre tipologie di audio legate alla velocità del paziente.

1.3 Seconda parte: Test su soggetti sani

Una volta ultimata la parte informatica del mio elaborato, ho svolto dei test su 20 soggetti per verificare se l'ausilio del feedback permettesse loro di svolgere gli esercizi in modo migliore.

Ciascun soggetto ha eseguito un test completo costituito da una serie di 6 task (Task A, Task B, Task C, Task D, Task E, Task F) diversi l'uno dall'altro.

Per rendere questo test il più attendibile e omogeneo possibile è stato scelto di randomizzare l'ordine dei task per ciascun soggetto, in modo tale che ogni soggetto dovesse affrontare una serie di test diversa l'una dall'altra.

L'elaborazione dei dati ottenuti da questi test è ancora in corso. In questo progetto verranno illustrate solo delle conclusioni preliminari a riguardo.

Capitolo 2

Sintesi del suono

Il suono nel mio elaborato ricopre un ruolo molto importante. Il mio scopo infatti è proprio quello di dimostrare che, con un adeguato feedback audio, il compimento dell'esercizio riabilitativo dovrebbe essere portato a termine con un minore sforzo e con risultati più soddisfacenti.

Per il mio elaborato ho preso in considerazione, sostanzialmente, due tipologie diverse di feedback: una legata alla velocità del target e una legata all'errore di posizione del soggetto rispetto al target. Per ciascuno di questi feedback inoltre sono state sviluppate, grazie alla collaborazione del Prof. Federico Avanzini e del dottorando Simone Spagnol due versioni del suono: una spazializzato e una no.

Il feedback audio verrà generato dal software Pure Data sfruttando i dati che gli vengono trasmessi via UDP. PD utilizza un particolare protocollo di invio dati denominato OSC (Open Sound Control) di cui adesso illustrerò tutte le caratteristiche.

2.1 Protocollo OSC

L'Open Sound Control (OSC) è un protocollo per la comunicazione tra computer, sintetizzatori audio e altri dispositivi multimediali ottimizzato per la tecnologia di networking. Tale protocollo è stato ideato e sviluppato nel 1997 da Adrian Freed and Matt Wright all'UC Berkeley Center for New Music and Audio Technology (CNMAT).

L'OSC è stato un'evoluzione del MIDI (Musical Instrument Digital Interface), ovvero il protocollo standard, utilizzato fin dagli anni 80, per l'interazione degli strumenti musicali elettronici.

La pressante domanda di riduzione dei costi, maggiore affidabilità e maggiore accessibilità a tutti gli utenti ha portato l'OSC a svilupparsi a tal punto da eguagliare e poi superare l'uso del MIDI.

L'OSC, infatti, presenta molti vantaggi rispetto al MIDI tra i quali una maggiore velocità dei dati di trasmissione e una risoluzione notevolmente superiore rispetto agli 8 bit concessi da MIDI.

2.1.1 Sintassi

Tipi di dati fondamentali

INT32: rappresenta un valore integer con segno a 32 bit in complemento a 2 memorizzati in formato big endian.

OSC-TIMETAG: numero a virgola fissa a 64 bit memorizzati in formato big endian.

FLOAT32: numeri a virgola mobile a 32 bit rappresentabili in formato IEEE754 memorizzati secondo il formato big endian.

STRINGA: sequenza di caratteri composta da byte nel caso delle stringhe con codifica ASCII alla cui fine è posto un byte di 0 per rendere il numero totale di bit multiplo di 32 e quindi il numero di byte multiplo di 4.

OSC-BLOB: un int32 che rappresenta la dimensione, seguito da un numero arbitrario di byte di dati in formato binario, seguiti a loro volta da 0-3 byte di zero per rendere il numero totale di bit multiplo di 32.

2.1.2 OSC Packets

I packets (più comunemente noti come pacchetti) sono l'unità fondamentale di trasmissione del protocollo OSC. L'applicazione che invia questi pacchetti prende il nome di client mentre quella che li riceve viene definita server.

Tali pacchetti sono costituiti sia da dati in essi contenuti, scritti in formato binario, sia dalla dimensione di questi dati. Come già specificato precedentemente,

il numero di byte che occupano deve essere multiplo di 4. Altro elemento molto importante da ricordare è che la rete, nell'atto di trasportare i pacchetti, deve tenere conto sia delle informazioni relative al contenuto sia di quelle relative alla dimensione che devono essere entrambe trasmesse.

I dati che vengono trasmessi in un pacchetto, riferito al protocollo OSC, si dividono in due tipologie: messaggio (message) o fascio (bundle). Il primo byte di dati permette di definire se il contenuto del pacchetto è del primo o del secondo tipo.

OSC TypeTag	OSC Arguments
i	int32
f	float32
s	stringa
b	OSC-blob

Figura 2.1: OSC Type Tag

OSC Messages

Un OSC Message è costituito da un OSC Address Pattern, ovvero una stringa che comincia con il carattere “/”, seguito da un OSC Type Tag String e da eventuali altri OSC Arguments.

Un OSC Type Tag String è una stringa che comincia con il carattere “,”, seguito da una serie di caratteri che rappresentano il tipo di dato OSC Arguments (visibile nella Tabella 3.1) a cui corrispondono.

Qualsiasi altra stringa contenente caratteri non riconosciuti dal protocollo OSC viene automaticamente scartata dal sistema.

OSC Arguments

Una sequenza di OSC Arguments viene identificata dalla serie di rappresentazioni in codice binario di ogni singolo argomento.

OSC Bundles

Un “bundle” (o fascio) è formato dalla stringa “#bundle” seguita da una Time Tag a cui seguono zero o più Bundle Elements. La Time Tag è una timetag a 64 bit a virgola fissa.

Un Bundle Element consiste della sua dimensione e del suo contenuto: la dimensione viene indicata da un int32, multiplo di 4, che rappresenta il numero di byte del contenuto, mentre il contenuto può essere un Message o un altro Bundle.

2.1.3 OSC Semantics

OSC Address e OSC Address Spaces

Ogni OSC server, ovvero l'applicazione che riceve il pacchetto di dati, possiede un set di metodi (OSC Methods). I metodi sono le potenziali destinazioni degli OSC Messages ricevuti dal server e corrispondono ad ognuno dei punti di controllo che l'applicazione mette a disposizione. L'invocazione di un OSC method corrisponde molto verosimilmente ad una chiamata a procedura. Vengono forniti gli argomenti al metodo per poi eseguirlo.

I metodi del server sono organizzati secondo una struttura dati dinamica ad albero chiamata OSC Address Space. Le foglie dell'OSC Address Space corrispondono ai metodi, mentre i rami corrispondono agli OSC Containers. Ogni Method ed ogni Container, eccetto il "root" (ovvero la radice), hanno un nome definito da una stringa in codifica ASCII di cui però non possono far parte i caratteri definiti nella tabella sottostante.

Carattere	Codice ASCII
"	32
#	35
*	42
,	44
/	47
?	63
[91
]	93
{	123
}	125

Figura 2.2: Caratteri non utilizzabili

L'OSC Address è un nome simbolico dato da un metodo per identificarlo all'interno della struttura dell'albero. Partendo dalla radice si separa ogni ramo che si incontra, ovvero ogni OSC Container, con il carattere "/" fino a giungere

al metodo interessato, il cui nome deve essere anche esso riportato. È stata scelta questa sintassi perché è la medesima scelta per gli URL.

Invio dei messaggi e pattern matching

Quando un server riceve un messaggio, deve invocare il metodo più appropriato per accedervi, basandosi essenzialmente sull'indirizzo del messaggio. Questo processo viene denominato *dispatching*. Il messaggio, una volta ricevuto dal server, deve essere inviato a tutti i metodi il cui indirizzo corrisponde a quello del messaggio. In poche parole il messaggio è inviato a tutti i metodi il cui OSC Address corrisponde all'OSC Address Pattern del messaggio stesso. La corrispondenza tra i due indirizzi si ha quando essi contengono lo stesso numero di parti e se tutte queste parti corrispondono carattere per carattere. In tale contesto si distinguono dei caratteri speciali:

- “?”: nell'OSC Address Pattern corrisponde positivamente ad ogni carattere dell'Address del metodo.
- “*”: nell'OSC Address Pattern corrisponde positivamente ad ogni 0 o sequenza di zero dell'Address del metodo.
- “[stringa di caratteri]”: nell'OSC Address Pattern corrisponde positivamente ad ogni carattere della stringa. All'interno della parentesi i caratteri “-” e “!” assumono significati diversi.
- Una lista di stringhe chiuse tra parentesi graffe e separate tra loro da una virgola (e.g. “{stringa1 , stringa2}”).

Temporal Semantics e OSC Time Tags

Il protocollo OSC non prevede alcun tipo di clock, al contrario invece un server OSC necessita di qualche meccanismo simile ad un clock. Nel caso in cui il server riceva un pacchetto contenente un unico OSC Message, esso deve solamente invocare il corrispondente metodo nel minor tempo possibile. Se invece il pacchetto dovesse contenere un unico OSC Bundle, risulta necessario un sistema che scandisca l'andamento temporale per sincronizzare ogni elemento dell'OSC Bundle

con il corrispondente metodo. Questo sistema di scansione temporale è costituito dalle Time Tags. Si prospettano due casi:

1. Time Tag minore o uguale al tempo corrente: il server deve invocare immediatamente il metodo a meno che l'utente non abbia impostato il server stesso in modo che rifiuti i messaggi che arrivano troppo tardi.
2. Time Tag maggiore del tempo corrente: le Time Tags possono rappresentare un tempo nel futuro; il server dovrebbe salvare l'OSC Bundle fino al tempo specificato ed invocare il metodo corretto.

Le Time Tags sono rappresentate da un numero a 64 bit a virgola fissa. I primi 32 bit rappresentano il numero di secondi dal 1/1/1990, mentre gli altri 32 bit rappresentano frazioni di secondi, fino ad una precisione di 200 ps. La Time Tag rappresentata da una serie di zero ed un 1 nel bit meno significativo identifica il fatto che il metodo debba essere invocato immediatamente.

In ultima analisi, andiamo ad evidenziare alcuni casi particolari:

- Quando un OSC Address Pattern corrisponde all'Address di più di un metodo, l'ordine con cui i metodi vengono invocati non ha importanza.
- Quando un OSC Bundle contiene più di un messaggio, i set di metodi corrispondenti ai vari messaggi devono essere invocati nello stesso ordine con cui i messaggi compaiono nel pacchetto.
- Quando un OSC Bundle contiene altri Bundle, la Time Tag del contenuto deve ovviamente essere maggiore di quella del contenitore in quanto quest'ultimo viene analizzato per primo.

2.2 Il nostro caso

Dopo aver illustrato come deve essere espressa la sintassi dei protocolli OSC per l'invio dei dati ad un'applicazione esterna, vado ora ad affrontare il mio particolare caso in cui invio un pacchetto dati al software Pure Data.

Prima di tutto è necessario definire il dato da inviare. Nel mio caso è costituito da un vettore di dimensioni 1x9:

```
pd_packet = [single(xy_pat(1)) single(xy_pat(2)) single(xtar(indt))
single(ytar(indt)) single(vtar_x(indt)) single(vtar_y(indt)) single(vpat_x)
single(vpat_y) single(flag)];
```

Dove:

- $xy_pat(1)$ e $xy_pat(2)$: indicano rispettivamente la posizione del soggetto rispetto all'asse x e rispetto all'asse y.
- $xtar(indt)$ e $ytar(indt)$: indicano rispettivamente la posizione del target rispetto all'asse x e rispetto all'asse y.
- $vtar_x(indt)$ e $vtar_y(indt)$: indicano rispettivamente la velocità del target rispetto all'asse x e rispetto all'asse y.
- $vpat_x(indt)$ e $vpat_y(indt)$: indicano rispettivamente la velocità del soggetto rispetto all'asse x e rispetto all'asse y.
- $flag$: indica il riferimento al feedback audio percepito dal paziente
- $indt$: è l'indice riferito al tempo calcolato con la funzione tic toc in Matlab.
- $single$: rappresenta la conversione necessaria a numeri a singola precisione. Tali numeri hanno una codifica a 32 bit che rispetta la dichiarazione di tipo float a 32 bit dell'OSC Message.

Successivamente si utilizza il metodo:

```
[attr datastring]=cstruct(pd_packet);
```

per scomporre il vettore `pd_packet` in un vettore di `unit8`. Tale vettore è la rappresentazione in virgola mobile del dato trasmesso in modalità, ovvero per primo viene passato il byte più significativo. Un dato di tipo `single` (formato da 4 byte) viene scomposto in un vettore di 4 `unit8`, i quali rappresentano i suoi byte. Il valore `datastring` rappresenta il vettore, mentre `attr` è una struttura che contiene i campi `size` (quanti byte sono richiesti per salvare il dato) e `align` (allineamento dei byte). I dati vengono convertiti in `single` prima di essere passati a `cstruct` in

quanto in questo modo viene rispettato l'OSC Arguments che richiede un numero a virgola mobile a 32 bit e l'allineamento a 4 byte.

Lo stesso metodo viene utilizzato per predisporre i byte di 0 necessari per rendere sempre in ogni caso il numero totale dei byte multiplo di 4. La stringa da inviare a Pure Data, ovvero l'OSC Server, è la seguente:

```
string=char(['/pd_packet' zerostring(1:2) ',fffffff' zerostring(1:2) datastring]);
```

dove le 9 "f" indicano i 9 float che rappresentano i valori contenuti nel pd_packet.

2.3 Pure Data

Pure Data (o Pd) è una piattaforma per la sintesi e l'elaborazione di audio, video e processi grafici in tempo reale.

Ideato e sviluppato da Miller S. Puckette negli anni 90, Pd è comunemente usato per creare spettacoli di musica dal vivo, effetti sonori, composizioni musicali, analisi audio, interfacciamento con sensori, controllo di robot e anche per interagire con siti web.

L'unità funzionale di Pure Data è una scatola ("box"). Un programma viene realizzato collegando tra loro una serie di scatole fino a formare un vero diagramma in cui si ha un flusso di dati continuo. Lo stesso programma è sempre in esecuzione, non c'è alcun tipo di separazione tra la scrittura del programma e l'esecuzione del programma stesso.

Le potenzialità di questo open source sono dunque illimitate, basta pensare che la comunità di utenti e programmatori di Pd hanno creato funzioni aggiuntive (chiamate "esterne" o "librerie esterne") che sono utilizzate per un'ampia varietà di altri scopi, come l'elaborazione video, la riproduzione e lo streaming di brani MP3 o video Quicktime, la manipolazione e la visualizzazione di oggetti 3D e la modellazione virtuale di oggetti fisici.

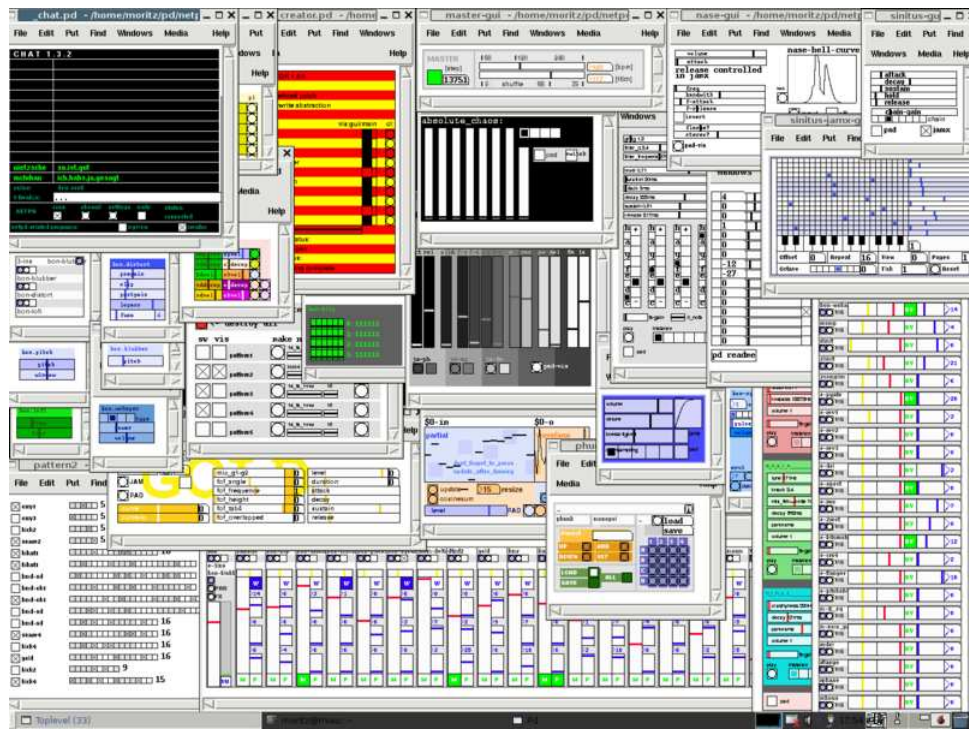


Figura 2.3: Esempio di interfaccia grafica in Pure Data

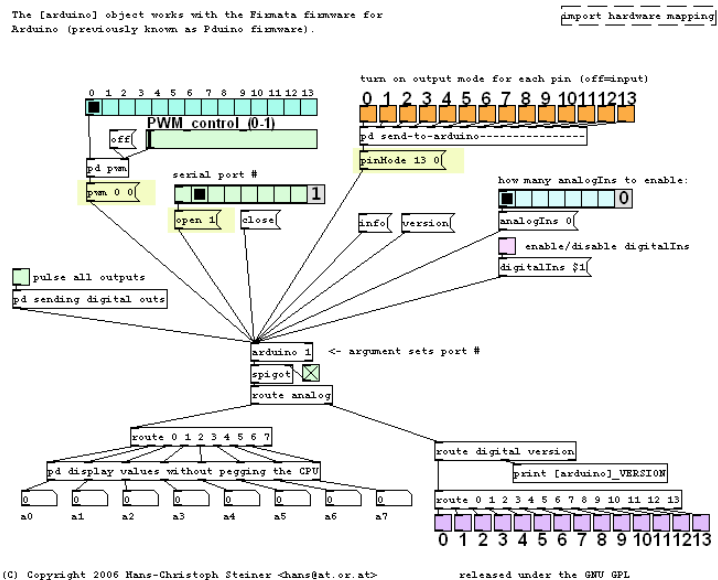


Figura 2.4: Esempio di struttura ad albero di Pure Data

2.3.1 Pure Data applicato alla mia interfaccia

Nel mio caso particolare, Pure Data è stato utilizzato per generare il feedback audio che avrebbero dovuto sentire i soggetti durante l'esecuzione dei test.

Pd in questo frangente ha svolto dunque una funzione di server tra la mia interfaccia GUI e l'audio che doveva essere generato. A Pure Data ho inviato un pacchetto di dati contenenti il flag, ovvero il tipo di audio selezionato (beep, velocità del target o errore di posizione) e le componenti in x e y di:

- Posizione del target e del soggetto
- Velocità target e velocità soggetto.

Bisogna sottolineare l'importanza del parametro flag, il quale permette di cambiare istantaneamente la selezione audio durante l'esercizio. Ad esempio, all'inizio dell'esercizio il flag è posto a 0 mentre subito prima di partire assume per 3 volte (di durata 0.5 s) il valore 9 che identifica il suono di un beep.

Dopo questi 3 segnali acustici l'esercizio ha inizio e il flag viene impostato a seconda della traccia audio che si vuole far sentire. Al termine dell'esercizio il flag torna a 0.

Flag	Modalità audio	Spazializzazione del suono
0	No audio	
1	Audio legato alla velocità del target	
2	Audio legato alla velocità del target	✓
3	Audio legato all'errore di posizione del paziente	
4	Audio legato all'errore di posizione del paziente	✓
5	/vuoto	
6	/vuoto	
7	/vuoto	
8	beep di fine esercizio	
9	beep di inizio esercizio	

Tabella 2.1: Tabella esplicativa dei vari flag associati alle diverse modalità di audio

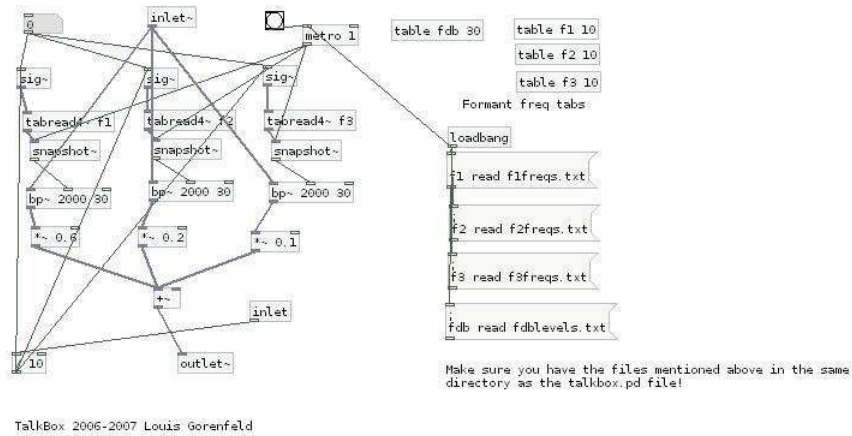


Figura 2.5: Finestra di talk box che mette in relazione tra loro i vari feedback audio

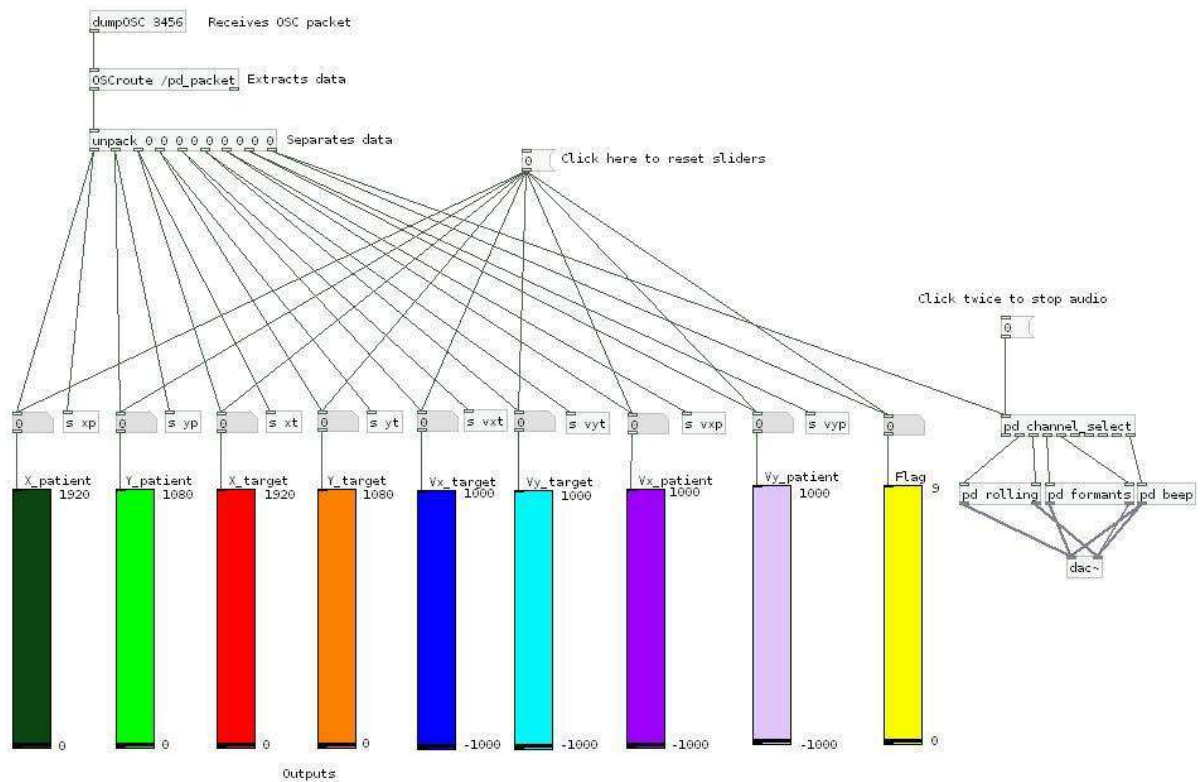


Figura 2.6: Interfaccia di Pure Data utilizzata per testare il funzionamento della GUI.

Capitolo 3

Software sviluppato

Il software sviluppato ha lo scopo di riprodurre un esercizio di un task motorio legato ad un target in movimento. Il paziente (“pallino verde”) dovrà quindi inseguire il target identificato dal “pallino rosso” che si muove con un movimento “punto-punto” lungo una traiettoria orizzontale a lunghezza fissa o a lunghezza variabile. Questo tipo di esercizio riabilitativo si basa sull’esperienza già realizzata in California in relazione all’esoscheletro Pneu-WREX.

Oltre al feedback visivo, il paziente avrà a disposizione anche un feedback audio implementato in due diverse modalità:

1. Legata alla velocità del target
2. Legata all’errore di posizione del paziente rispetto al target.



Figura 3.1: Esempio di esecuzione del task motorio considerato

3.1 Implementazione interfaccia

In quest'ambito di ricerca è stata realizzata una GUI di Matlab (Figura 3.1) che prevedesse due modalità d'esecuzione:

1. Modalità online: dove si visualizzava alla frequenza di 50 Hz i dati relativi agli spostamenti del target e del soggetto, lungo gli assi x e y (sulla tavoletta grafica), e contemporaneamente tali dati venivano inviati via UDP a PureData in modo che questo software potesse produrre il feedback audio percepito dal soggetto durante l'esecuzione del test. In questa modalità è possibile inoltre salvare in un file .mat i dati riguardanti i parametri del sistema e le performance del soggetto in una struttura dati di Matlab.
2. Modalità offline: dove si possono rivedere le performance di ciascun soggetto (sempre alla frequenza di 50 Hz) su una finestra di dimensioni minori.

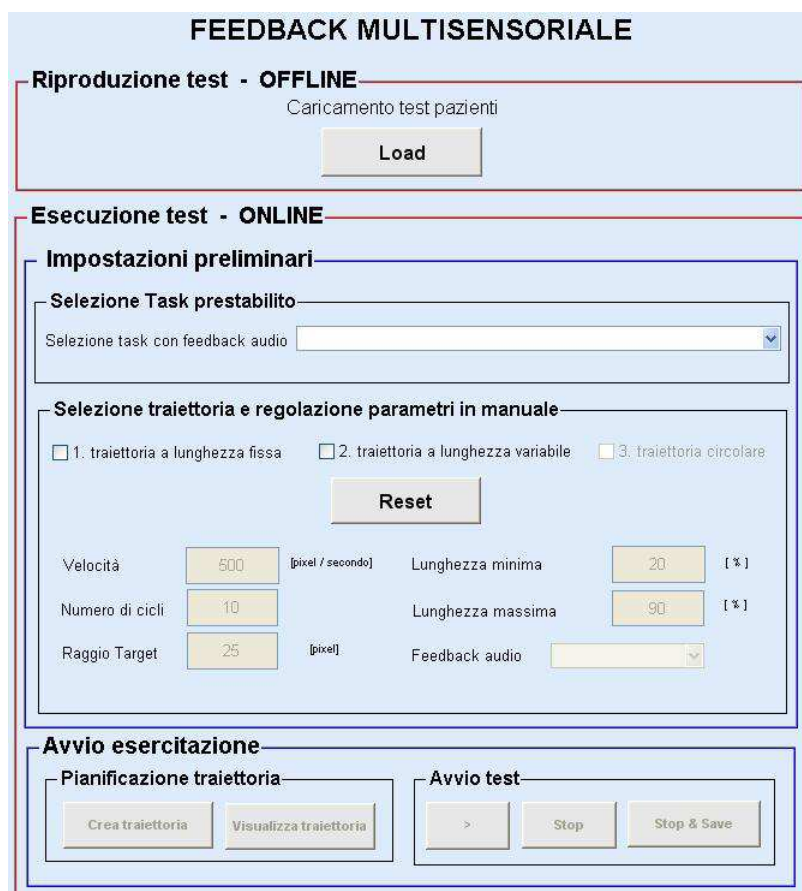


Figura 3.2: Finestra dell'Interfaccia che compare all'avvio di Matlab

3.2 Istruzioni preliminari per l'utilizzo dell'interfaccia

All'utente, una volta lanciato il software che genera l'interfaccia, si presenta una doppia scelta:

1. “Riproduzione test – OFFLINE”: permette di caricare i dati preesistenti di un determinato soggetto e rivedere l'esecuzione dell'esercizio stesso.
2. “Esecuzione test – ONLINE”: permette di far eseguire un nuovo esercizio scegliendo tra due diverse modalità:

Per quanto riguarda la modalità ONLINE si individuano due diverse tipologie di impostazione dell'esercizio:

- “Selezione Task Prestabilito”: carica dei parametri prestabiliti per l'esercizio.
- “Selezione Traiettoria e regolazione parametri in manuale”: dà la possibilità di creare un esercizio a seconda delle proprie esigenze gestendo manualmente l'impostazione di ciascun singolo parametro.

Inoltre, ogni volta che si apre l'interfaccia, compare nel display secondario, o nello schermo principale (se si ha a disposizione un solo schermo), la finestra (Fig. 3.3) dove verrà svolto l'esercizio da parte del soggetto con il target già posizionato al centro schermo.

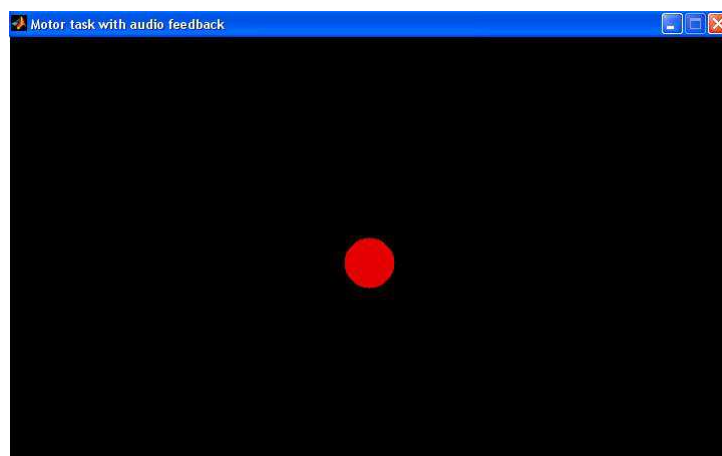


Figura 3.3: Finestra di esecuzione dell'esercizio, con il target visibile al centro

3.2.1 Istruzioni: “ESECUZIONE TEST - ONLINE”

Partiamo ad analizzare il caso che si voglia far svolgere un esercizio ad un soggetto. All’utente si prospettano due possibilità: selezionare manualmente una traiettoria o eseguire un Task.

“Selezione traiettoria e regolazione dei parametri in manuale”

Permette di gestire in modo manuale tutti i parametri messi a disposizione dalla GUI.



The screenshot shows a window titled "Selezione traiettoria e regolazione parametri in manuale". At the top, there are three radio buttons for trajectory selection: "1. traiettoria a lunghezza fissa" (unchecked), "2. traiettoria a lunghezza variabile" (checked), and "3. traiettoria circolare" (unchecked). Below the radio buttons is a "Reset" button. The main area contains several input fields: "Velocità" (500 [pixel / secondo]), "Numero di cicli" (10), "Raggio Target" (25 [pixel]), "Lunghezza minima" (20 [%]), "Lunghezza massima" (90 [%]), and "Feedback audio" (a dropdown menu).

Figura 3.4: Riquadro selezione manuale di un esercizio

Il primo parametro da impostare è la scelta della traiettoria. Sono disponibili due tipi diversi di traiettoria:

- Traiettoria a lunghezza fissa, caratterizzata dalla ripetizione di tratti aventi tutti la stessa lunghezza.
- Traiettoria a lunghezza variabile, caratterizzata dalla ripetizione di tratti di lunghezza differente l’uno dall’altro.

Se l’utente dovesse compiere la scelta errata e, quindi, spuntare il “checkbox” non desiderato è possibile annullare la propria decisione premendo il pulsante “Reset” e tornare nuovamente alla scelta della traiettoria.

I successivi parametri regolabili sono:

- *Velocità*: indica la velocità del target espressa in pixel/secondo. Di default questo parametro ha il valore 500 perché, svolgendo alcuni test prelimi-

nari, abbiamo verificato che un valore superiore a questo rendeva molto difficoltoso l'inseguimento del target.

- *Numero cicli*: indica il numero di ripetizione dx-sx compiute dal target durante l'esecuzione dell'esercizio selezionato. Di default questo parametro ha il valore 10.
- *Raggio target*: indica il raggio del pallino rosso che dovrà essere inseguito durante l'esercizio. Di default questo parametro ha il valore 25, se lo si modifica e si preme il tasto "invio" si vedrà cambiare il target nella finestra "Motor task with audio feedback" con le nuove dimensioni impostate.
- *Lunghezza minima*: indica il limite minimo che il tratto compiuto dal target può compiere durante l'esecuzione dell'esercizio. Di default questo parametro assume il valore 0 nella "traiettoria a lunghezza fissa" mentre assume il valore 20 nella "traiettoria a lunghezza variabile".
- *Lunghezza massima*: indica il limite massimo che il tratto compiuto dal target può compiere durante l'esecuzione dell'esercizio. Di default questo parametro assume il valore 90 in entrambe le traiettorie.

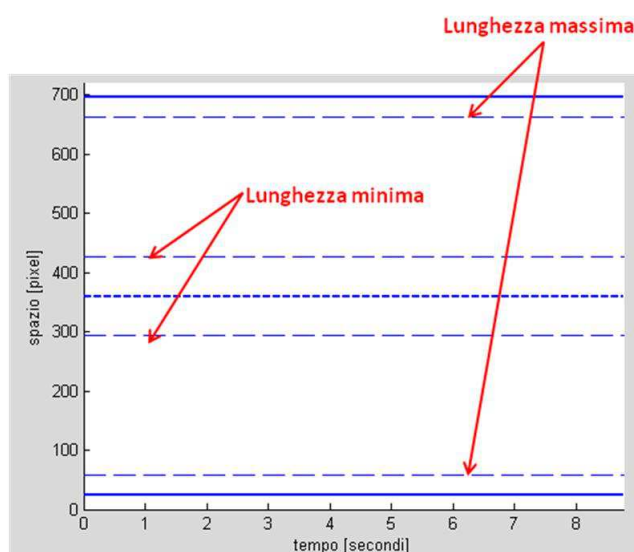


Figura 3.5: Visualizzazione nel grafico delle linee che individuano la lunghezza massima e la lunghezza minima in uno schermo unico

- *Feedback audio*: attraverso un menù a tendina si possono scegliere le 5 diverse tipologie di audio: una in cui l'audio non è presente, due (spazializzata o no) legate alla velocità del target e due (spazializzata o no) legate all'errore di posizione del soggetto.

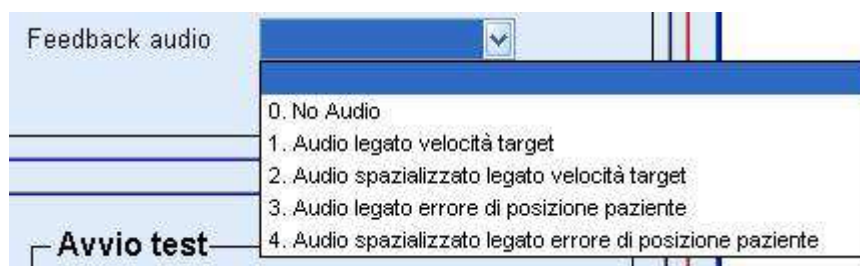


Figura 3.6: Riquadro selezione del feedback audio

“Selezione Task prestabilito”

Scegliere un task prestabilito significa scegliere tra 6 diverse tipologie di esercizio in cui i parametri sono già stati impostati automaticamente.



Figura 3.7: Riquadro selezione Task

- **TASK 0**: consiste in un test preliminare in cui il paziente è libero di muovere il proprio puntatore sulla tavoletta grafica per verificarne la sensibilità nei movimenti, mentre il target rimane fisso a centro-schermo.

- TASK A: consiste nell'esecuzione di un esercizio a *lunghezza fissa* in cui sono stati impostati i seguenti parametri:
 - Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 0 %
 - Lunghezza massima: 60%
 - Feedback audio: No audio (0)

- TASK B: consiste nell'esecuzione di un esercizio a *lunghezza variabile* in cui sono stati impostati i seguenti parametri:
 - Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 20 %
 - Lunghezza massima: 80 %
 - Feedback audio: No audio (0)

- TASK C: consiste nell'esecuzione di un esercizio a *lunghezza fissa* in cui sono stati impostati i seguenti parametri:
 - Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 0 %
 - Lunghezza massima: 60 %
 - Feedback audio: Audio spazializzato legato alla velocità del target (2)

- TASK D: consiste nell'esecuzione di un esercizio a *lunghezza variabile* in cui sono stati impostati i seguenti parametri:

- Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 20 %
 - Lunghezza massima: 80 %
 - Feedback audio: Audio spazializzato legato alla velocità del target (2)
- TASK E: consiste nell'esecuzione di un esercizio a *lunghezza fissa* in cui sono stati impostati i seguenti parametri:
 - Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 0 %
 - Lunghezza massima: 60 %
 - Feedback audio: Audio spazializzato legato all'errore di posizione del paziente (4)
- TASK F: consiste nell'esecuzione di un esercizio a *lunghezza variabile* in cui sono stati impostati i seguenti parametri:
 - Velocità: 400 pixel/secondo
 - Numero cicli: 28
 - Raggio Target: 25 pixel
 - Lunghezza minima: 20 %
 - Lunghezza massima: 80 %
 - Feedback audio: Audio spazializzato legato all'errore di posizione del paziente (4)

	Lunghezza costante	Lunghezza variabile	Feedback audio
0	/vuoto	/vuoto	0
A	✓		0
B		✓	0
C	✓		2
D		✓	2
E	✓		4
F		✓	4

Tabella 3.1: Sintesi delle caratteristiche dei Task

“Pianificazione traiettoria”

Ultimata la parte delle “Impostazioni preliminari” si passa alla creazione della traiettoria desiderata e alla sua possibile visualizzazione.

Il pulsante “Crea Traiettoria” realizza la traiettoria desiderata basandosi sui parametri precedentemente impostati dall’utente. Per creare tale traiettoria abbiamo utilizzato la funzione: `traj_pp`. Tale funzione pianifica la traiettoria di un moto punto per punto. Nel nostro caso essa si basa sull’interpolazione di un polinomio di 5° grado con velocità e accelerazione nulle negli istanti iniziali e finali del moto.

$$\mathbf{x}(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3 + \mathbf{a}_4 t^4 + \mathbf{a}_5 t^5$$

$$\mathbf{x}(0) = \mathbf{x}_{in} \quad \mathbf{x}(T) = \mathbf{x}_{fin}$$

$$\dot{\mathbf{x}}(0) = \mathbf{0} \quad \dot{\mathbf{x}}(T) = \mathbf{0}$$

$$\ddot{\mathbf{x}}(0) = \mathbf{0} \quad \ddot{\mathbf{x}}(T) = \mathbf{0}$$

Figura 3.8: Equazione di un polinomio di 5° grado che esprime la pianificazione della traiettoria considerata per il progetto

Il pulsante “Crea Traiettoria” viene disabilitato nel caso “Selezione Task Prestabilito”. In questo caso si può solo visualizzare la traiettoria creata al momento della selezione.

Il pulsante “Visualizza Traiettoria” permette di visualizzare sullo schermo la traiettoria creata manualmente o tramite la scelta del Task. Nel grafico che si apre

premendo il pulsante viene rappresentato l'andamento della traiettoria rispetto all'asse dei tempi (ascissa) e alle dimensioni dello schermo (ordinate).

Nel caso della selezione manuale dei parametri (nel caso di lunghezza variabile) se la traiettoria visualizzata non dovesse risultare soddisfacente alle esigenze dell'utente si può premere nuovamente il pulsante “Crea Traiettoria” per generare una nuova traiettoria e visualizzarla.

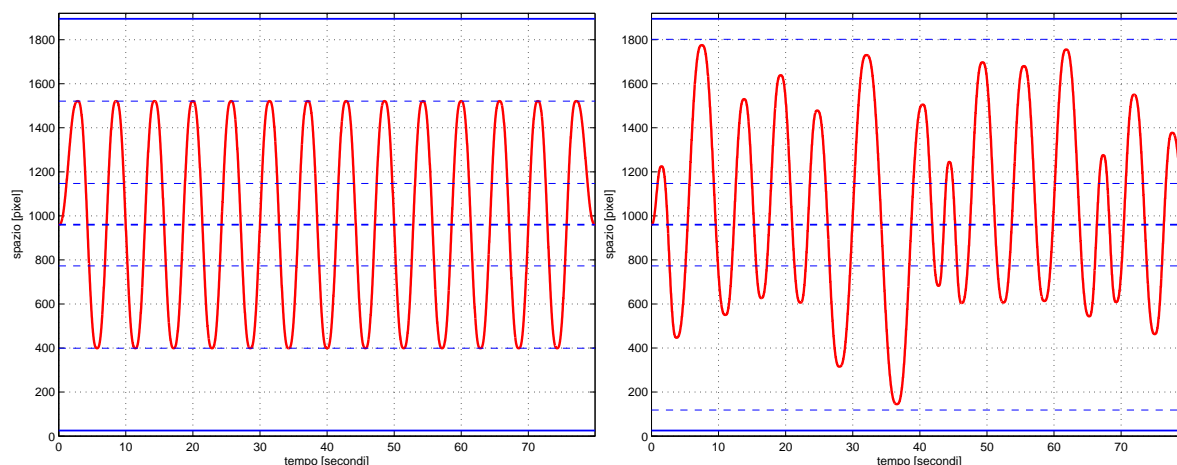


Figura 3.9: Grafici ottenuti dalla pianificazione di una traiettoria a lunghezza fissa e di una a lunghezza variabile

“Avvio test”

Una volta impostata e creata la traiettoria, con i parametri scelti a proprio piacimento nelle “Impostazioni preliminari”, è ora possibile far partire l'esecuzione dell'esercizio dalla finestra “Avvio Test”.



Figura 3.10: Riquadro della sezione Avvio test

Come si può vedere dalla Figura, tale finestra è costituita da 3 pulsanti.

Inizialmente l'unico pulsante accessibile all'utente è quello dello start “>”. Andiamo ora ad analizzare singolarmente ciascun pulsante per illustrarne caratteristiche e funzioni:

- Pulsante “>”: dà inizio all’esercizio (task o esercizio impostato manualmente). Premuto il pulsante, la stringa che lo rappresenta nella visualizzazione diventa “II”. Una volta preso il via l’esercizio, può essere messo in pausa premendo lo stesso bottone.
- Pulsante “Stop”: interrompe l’esercizio escludendo la possibilità di riprenderlo da dove era stato interrotto. Il target viene riposizionato al centro dello schermo e il pulsante “Reset” ritorna abilitato.
- Pulsante “Stop & Save”: ha una funzione analoga al pulsante “Stop”, diversamente da quest’ultimo però esso permette il salvataggio di dati una volta interrotto o terminato l’esercizio. I dati verranno salvati sottoforma di una struct di Matlab, ovvero una struttura dati che può contenere diversi parametri (matrici o valori) ognuno abbinato al proprio nome. Nel nostro caso la struttura dati è stata definita “exercise_parameters” e comprende i seguenti valori:
 - Una matrice denominata “data” contenente 12 colonne e un numero di righe pari alla lunghezza dei punti della traiettoria scelta. Ciascuna colonna rappresenta in ordine: il tempo corrispondente ai toc del timer (“t_att”), le posizioni in x e in y del paziente (“xy_pat(1)” e “xy_pat(2)”), le posizioni in x e in y del target (“x_tar” e “y_tar”), le velocità in x e in y del target (“vtar_x” e “vtar_y”), le velocità in x e in y del paziente (“vpat_x” e “vpat_y”), un contatore di quante volte si preme il tasto play (“play_counter”), un indicatore della modalità di feedback in esecuzione (“flag”) e un contatore che identifica in quale tratto si trovano paziente e target (“tratto”).
 - 9 parametri corrispondenti a: velocità del target, numero delle iterazioni, lunghezza minima, lunghezza massima, raggio del target, dimensioni dello schermo, traiettoria selezionata, stringa contenente una breve descrizione dell’esercizio e un parametro che il grado del polinomio con il quale è stata costruita la traiettoria scelta (nel nostro caso sarà sempre pari a 5). La struttura dati potrà essere salvata, attraverso la finestra mostrata in Figura 3.1.2.3, sottoforma di un file

.mat nella cartella “dati pazienti” come nel nostro caso o in qualsiasi altra cartella scelta dall’utente.

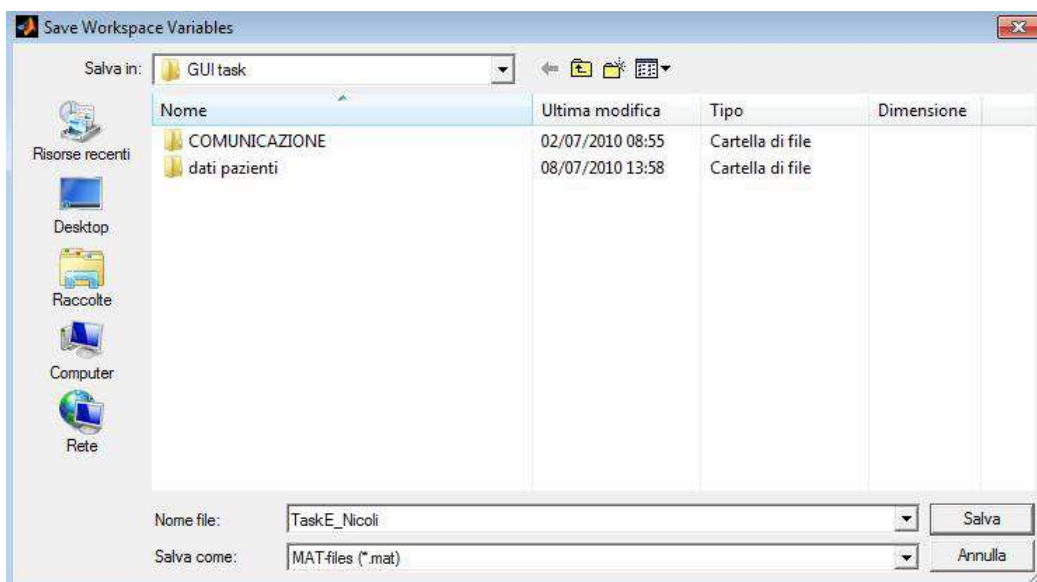


Figura 3.11: Finestra di salvataggio dati

“Impostazioni del Timer”

Per quanto riguarda la visualizzazione dei “pallini” del target e del paziente sullo schermo ci siamo avvalsi della funzione timer in Matlab. Più precisamente abbiamo realizzato una funzione da noi denominata draw, associata alla proprietà TimerFcn. Ricordiamo inoltre che la frequenza caratteristica del timer è stata impostata a 50 Hz. La funzione draw è suddivisa in 2 sezioni distinte:

1. dedicata alla modalità di esercizio eseguito in real-time (“Esecuzione test – ONLINE”)
2. dedicata alla modalità di riproduzione dell’esercizio (“Esecuzione test – OFFLINE”).

E’ il valore del parametro isLoad, definito all’interno della funzione draw, che stabilisce in quale modalità di esecuzione ci troviamo:

- isLoad pari a 1 indica che è in atto la modalità con esercizio caricato da memoria

- isLoad pari a 0 indica che l'esercizio sta avvenendo in tempo reale.

Per quanto riguarda l'esercizio in real-time, ovvero la modalità "ON-LINE", verranno distinte 3 fasi:

1. fase pre-esercizio
2. fase di esercizio
3. fase post-esercizio

In ognuna di queste tre fasi la funzione draw esegue solo alcune delle operazioni fondamentali per cui è predisposta, tra le quali:

- visualizzazione a video della posizione del soggetto e del target
- caricamento dei dati relativi al tempo della traiettoria, posizione e velocità del soggetto e target, e altri dati relativi al tipo di esercizio in una matrice da utilizzare per un'eventuale memorizzazione dell'esercizio
- invio di un pacchetto dati ad un software di elaborazione (nel nostro caso Pure Data) per la generazione di un feedback sonoro.

Per quanto riguarda la visualizzazione sullo schermo dell'andamento dell'esercizio, la posizione del soggetto è estrapolata da una funzione che restituisce le coordinate del puntatore (mouse o nel nostro caso il "pennino" della tavoletta grafica). Le coordinate assolute restituite dal puntatore vengono ricondotte a coordinate relative dello schermo su cui è visualizzato l'esercizio tramite una sottrazione per eliminare l'offset sulle coordinate dovuto allo schermo secondario.

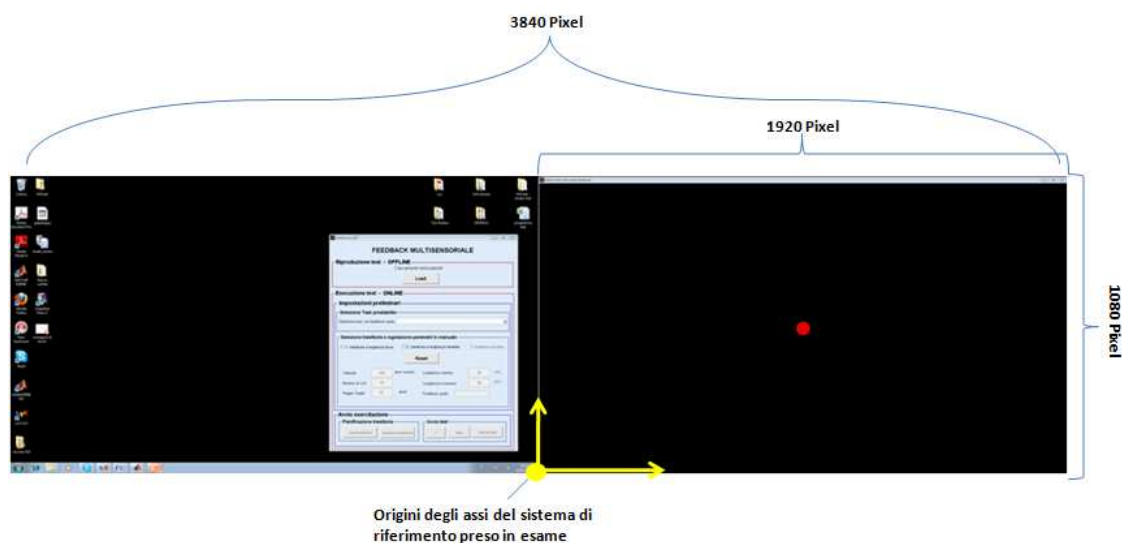


Figura 3.12: Rappresentazione grafica delle dimensioni degli schermi e relativa origine degli assi di riferimento presi in considerazione per moto del target e del paziente

Per il tracciamento della posizione del target la funzione draw si serve del vettore dei tempi della traiettoria, del vettore della traiettoria (in x e in y) e di un cronometro gestito attraverso i comandi tic (azzeramento e start), toc (intertempo). L'intertempo del cronometro, avviato in concomitanza con l'inizio del task, viene utilizzato per campionare opportunamente il vettore dei tempi della traiettoria.

Così la traiettoria precedentemente creata viene disegnata in modo coerente con il tempo che è realmente trascorso dall'avvio dell'esercizio, e non con il tempo che l'oggetto timer usa per invocare la sua funzione TimerFcn il quale risente dei ritardi dovuti a tutte le operazioni che devono essere eseguite ad ogni sua chiamata.

Ogni posizione che viene visualizzata viene anche memorizzata in una matrice che consentirà la riproduzione in tempi successivi dello stesso esercizio. Un contatore controlla la riga di salvataggio della matrice mentre nelle colonne si salvano:

1. istante temporale t
2. x del soggetto all'istante t

3. y del soggetto all'istante t
4. x del target all'istante t
5. y del target all'istante t
6. velocità in x del target all'istante t
7. velocità in y del target all'istante t
8. velocità in x del soggetto all'istante t
9. velocità in y del soggetto all'istante t
10. numero di riprese dell'esercizio
11. indicatore del tipo di audio associato all'esercizio
12. numero del tratto

Queste prime due operazioni fondamentali (visualizzazione e memorizzazione) vengono eseguite solo ed esclusivamente nella fase di esercizio. La comunicazione con Pd, avviene in tutte le 3 fasi, ovviamente con modalità diverse segnalate dal valore del parametro *flag*.

Nella fase di pre-esercizio (di durata di 5 secondi) si fa uso della comunicazione con Pure Data, per generare una successione di 3 “beep”, una sorta di “countdown” dal terzo al quinto secondo, per segnalare al soggetto l'imminenza dell'avvio del Task. I “beep” sono controllati specificando nel campo *flag* del pacchetto UDP (che per comodità ha sempre la stessa struttura anche se molti campi non sono indispensabili in tutte le 3 fasi) una successione di 0 e 9 (con cambio ogni 0.5 secondi) per attivare e disattivare il “beep” (*flag* = 0 indica assenza di audio).

Nella fase di esercizio il pacchetto UDP che si invia a PureData è nella sua forma completa, ovvero tutti i campi sono riempiti con i valori corretti che il software utilizza per generare un feedback sonoro del tipo specificato dal valore del *flag*:

1. audio legato alla velocità del target non spazializzato
2. audio legato alla velocità del target spazializzato

3. audio legato all'errore di posizione non spazializzato

4. audio legato all'errore di posizione spazializzato

Nel pacchetto inviato in questa fase risultano quindi fondamentali anche i campi contenenti posizione in x e y del soggetto e del target, velocità in x e y del soggetto e del target, che verranno pertanto completati opportunamente per consentire una corretta generazione del suono.

Nella fase di post esercizio i dati relativi alla posizione e alla velocità del target e del soggetto vengono “congelati” in quelli relativi all'ultimo campione dell'esercizio mentre si utilizza il flag, impostato al valore 8, per generare un beep di fine esercizio della durata di 0.5 secondi.

Ricordiamo il significato dei vari *flag*:

- no audio (0)
- audio legato alla velocità del target non spazializzato (1)
- audio legato alla velocità del target spazializzato (2)
- audio legato all'errore di posizione non spazializzato (3)
- audio legato all'errore di posizione spazializzato (4)
- /vuoto (5)
- /vuoto (6)
- /vuoto (7)
- beep di fine esercizio (8)
- beep di countdown (inizio esercizio) (9)

3.2.2 Visualizzazione ed invio dei dati a Pure Data

PNET

La “pnet” è una particolare funzione I/O realizzata per la comunicazione via TCP / UDP in Matlab sviluppata da Peter Rydesater della Mid Sweden University.

Questa funzione può essere chiamata con diverse opzioni/comandi ed è in grado di supportare il trasferimento di dati di diverse tipologie e ordini di byte.

Differenze tra UDP e TCP

Nel mio elaborato è stato scelto di utilizzare il protocollo UDP (User Datagram Protocol) piuttosto che il protocollo TCP (Transmission Control Protocol) per la comunicazione dei dati tra Matlab e Pure Data. Le principali differenze tra questi due protocolli sono:

1. l'UDP è un protocollo di tipo connectionless, ovvero suddivide il flusso di dati in entità (pacchetti) che vengono instradati in modo indipendente l'uno dall'altro, senza interazioni di ritorno tra sorgente e destinatario. L'UDP quindi non gestisce il riordinamento dei pacchetti né la ritrasmissione di quelli persi, ed è perciò generalmente considerato di minore affidabilità;
2. il TCP garantisce il trasferimento affidabile dei dati, il controllo di flusso e il controllo della congestione.
3. l'UDP fornisce soltanto i servizi basilari del livello di trasporto, ovvero:
 - (a) moltiplicazione delle connessioni, ottenuta attraverso il meccanismo delle porte
 - (b) verifica degli errori mediante una checksum, inserita in un campo dell'intestazione del pacchetto.

Riassumendo, il protocollo UDP è un protocollo stateless, ovvero non tiene nota dello stato della connessione, dunque ha rispetto al TCP informazioni in meno da memorizzare ed inoltre è molto rapido ed efficiente per le applicazioni "leggere", come nel nostro caso.

Il nostro caso Per la comunicazione via UDP tra l'interfaccia realizzata e PureData ho utilizzato alcuni metodi messi a disposizione dalla pnet. Innanzitutto ho creato il socket per la comunicazione dati e ho impostato la porta sorgente:

```
mysock = pnet('udpsocket', 3455);
```

Dove “udpsocket” indica che il socket creato è per la comunicazione dati via UDP mentre 3455 è il valore che identifica la porta sorgente. Successivamente bisogna impostare l’indirizzo IP e la porta di destinazione, identificati grazie al software SocketSniff.

```
pnet(mysock, 'udpconnect', '127.0.0.1', 3456);
```

In questo caso il valore 3456 identifica la porta remota che riceve il socket. Dopo aver ultimato il dato da inviare, sempre in accordo al protocollo OSC, ho potuto scrivere il dato nel buffer del socket e inviarlo.

```
pnet(sock, 'write', string);
```

```
pnet(sock, 'writepacket');
```

3.2.3 Istruzioni: “RIPRODUZIONE TEST - OFFLINE”

Nell’ interfaccia è stata implementata anche la possibilità di rivedere la performance di ciascun soggetto attraverso il caricamento dei suoi dati per mezzo del pulsante *Load*.

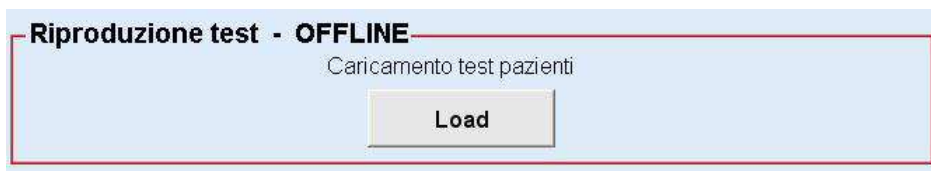


Figura 3.13: Riquadro del pulsante Load

Cliccando su questo pulsante comparirà una finestra in cui sono contenuti tutti i dati salvati dei diversi soggetti.

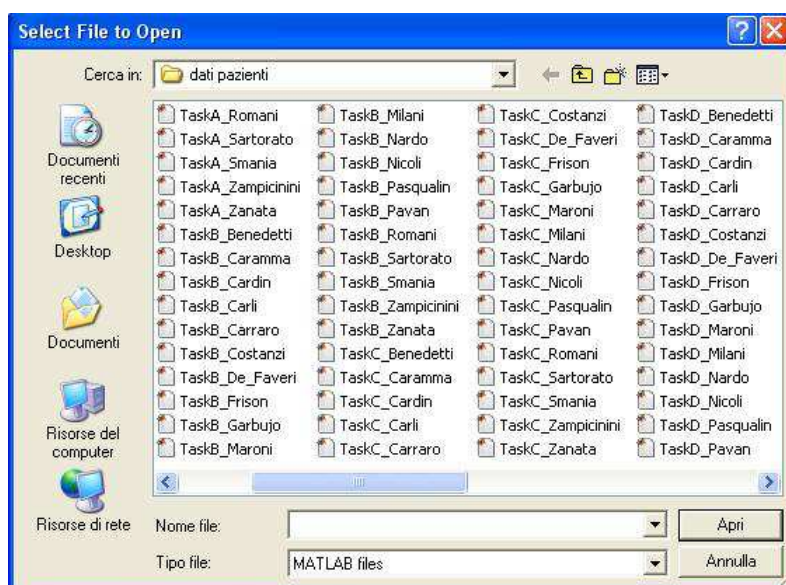


Figura 3.14: Finestra di caricamento dati

Effettuata la scelta, vengono caricati in automatico tutti i parametri iniziali e le posizioni (in x e in y), del target e del soggetto stesso, relative all'esercizio svolto dal soggetto desiderato. Si potrà quindi rivedere il grafico della traiettoria svolta e la riproduzione dell'esercizio stesso. La finestra in cui verrà riprodotto l'esercizio svolto comparirà nello schermo primario (quello in cui è presente l'interfaccia) con dimensioni scalate del 50%.

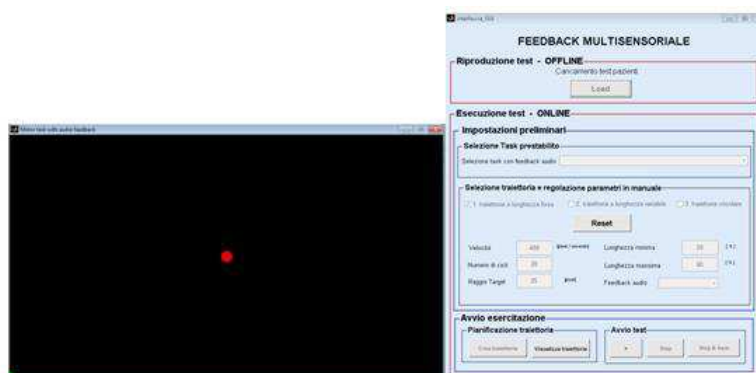


Figura 3.15: Visualizzazione del desktop in modalità OFFLINE

Oltre alla riproduzione visiva sarà possibile anche sentire il feedback percepito dal soggetto in ogni istante della sua esecuzione.

Questa modalità può essere molto utile al medico per capire in quali particolari tratti il soggetto ha trovato difficoltà e come ha risposto al feedback audio nelle diverse ripetizioni dei Task.

La possibilità di ricontrollare l'esercizio svolto da un soggetto, è stata molto utile anche per lo scopo del mio elaborato. Infatti la revisione di questi video ci ha permesso di escludere i soggetti che secondo noi si sono particolarmente distratti nello svolgimento della prova e per questo nella rielaborazione finale sono stati considerati degli outlier.

“Impostazioni del Timer”

Quando il valore del parametro `isLoad` è impostato a 1 si è in modalità riproduzione. L'esercizio è già stato eseguito e memorizzato, ci si limita soltanto a caricare in una struttura tutti i dati relativi all'esercizio che si vuol riprodurre e di rappresentarlo con i tempi giusti a video. Pertanto, a differenza della modalità on-line, ora abbiamo una sola fase cioè quella di visualizzazione: una volta avviato l'esercizio questo parte senza fasi preliminari e si arresta all'ultimo valore della traiettoria senza successive fasi conclusive. Semplificate sono anche le operazioni fondamentali che la funzione deve compiere in questa modalità. Esse si riducono a:

- visualizzazione a video istante per istante del corretto campione della posizione di soggetto e target scelto con l'ausilio del cronometro precedentemente spiegato, fra i vettori di una matrice contenente i dati di tutto l'esercizio che si è caricato.
- invio del pacchetto a Pure Data per la generazione del feedback sonoro come lo si era sentito durante l'esecuzione originale dell'esercizio

Nella modalità di riproduzione perciò si deve andare a “pescare”, coerentemente con il tempo di esercizio, all'interno di una struttura (`exercise_parameters`) contenente tutte le informazioni relative ad un esercizio precedentemente svolto, i dati necessari per la visualizzazione del task sullo schermo e di mantenere la comunicazione con Pure Data per rigenerare lo stesso feedback audio fornito nell'esecuzione originale.

Capitolo 4

Test sperimentali

4.1 Pianificazione

Ultimata l'implementazione dell'interfaccia GUI abbiamo provveduto ad organizzare due giornate di test per verificare se effettivamente l'utilizzo del feedback sonoro potesse agevolare l'esecuzione di questo tipo di esercizio riabilitativo. Come già evidenziato nel paragrafo 3.1.2.2, per questi test, abbiamo selezionato sei diverse tipologie di task:

1. Task A: traiettoria a lunghezza fissa priva di audio.
2. Task B: traiettoria a lunghezza variabile priva di audio.
3. Task C: traiettoria a lunghezza fissa con audio spazializzato legato alla velocità del target.
4. Task D: traiettoria a lunghezza variabile con audio spazializzato legato alla velocità del target.
5. Task E: traiettoria a lunghezza fissa con audio spazializzato legato all'errore di posizione del paziente rispetto al target.
6. Task F: traiettoria a lunghezza variabile con audio spazializzato legato all'errore di posizione del soggetto rispetto al target.

	Lunghezza costante	Lunghezza variabile	Feedback audio
0	/vuoto	/vuoto	0
A	✓		0
B		✓	0
C	✓		2
D		✓	2
E	✓		4
F		✓	4

Tabella 4.1: Sintesi delle caratteristiche dei Task

Per cercare di ottenere un risultato più veritiero possibile abbiamo selezionato un gruppo di soggetti omogeneo: 20 persone di età compresa fra i 21 e i 29 anni, di cui 11 uomini e 9 donne, tutti destrorsi.

Inoltre abbiamo randomizzato i task per ciascun soggetto in modo che l'ordine d'esecuzione degli esercizi fosse diverso per tutte le 20 persone. Per fare questo abbiamo realizzato una function in Matlab a cui venivano passati come parametri di ingresso il numero di soggetti e il numero di task da svolgere. Da questa veniva restituita una permutazione dei task per ciascun soggetto. Dai valori così ottenuti abbiamo realizzato una tabella in Excel, utilizzata durante i test per segnare la presenza del soggetto e particolari anomalie durante l'esecuzione dei task.

I test si svolsero dalle 14.00 del 7 luglio fino alle 15.00 dell'8 luglio 2010.

PIANIFICAZIONE TASKS - Generazione di feedback sonoro nell'esecuzione di task motori

SOGGETTO	TASK	ETA	SESSO	BRACCIO	IRREGOLARITA' NEI TEST				RIMA PRESENZA DEL SOGGETTO		
					Distrazione sogg.	Mano abasa	Scatti pc			Interruzione task	
							si	nesso scatto	si	nessa interruzione	
1	C E B F A D										
2	E C B F A D										
3	D B C F A E										
4	E C F B A D										
5	B C A F E D										
6	C A E D B F										
7	F E D B C A										
8	D C E F B A										
9	A E B F D C										
10	F D B E A C										
08-lug											
11	B D F A C E										
12	C F E B D A										
13	F E D A C B										
14	D F B A E C										
15	F A B D C E										
16	C A F D E B										
17	B D C F A E										
18	E F C A D B										
19	E B C A F D										
20	D F A C E B										

Figura 4.1: Tabella base utilizzata nella sessione di test

A ciascun soggetto, prima di iniziare il test, venivano illustrate alcune norme di base per eseguire l'esperienza in modo corretto, tra le quali:

“Il soggetto PRIMA di cominciare il test dovrà svolgere le seguenti operazioni preliminari:

- 1. Posizionarsi sullo sgabello in modo da essere a distanza corretta dal tavolo e al tempo stesso avere una buona visuale del monitor.*
- 2. Assicurarci che il braccio con cui svolgerà l'esercizio non sia troppo teso o troppo rilassato.*
- 3. Assicurarci che le cuffie siano posizionate in modo corretto sulle orecchie.*
- 4. Posizionare la mano inutilizzata nell'esercizio sul bordo scuro della tavoletta grafica.*
- 5. Posizionarsi con la “penna” al centro della tavoletta grafica.*

DURANTE l'esercizio il soggetto dovrà:

- 1. Rimanere concentrato sul test che sta svolgendo fino alla fine di esso. Se per qualche ragione il test dovesse interrompersi una volta incominciato, sarà discrezione dei due testisti stopparlo e farlo ricominciare dall'inizio.*
- 2. Non alzare mai la penna dalla tavoletta grafica.*
- 3. Non muoversi sullo sgabello.*

Il test durerà 15 minuti circa: di cui 80 secondi per ciascun task (6 in totale), 30 secondi di pausa tra un task e l'altro e 60 secondi di prova iniziale con la tavoletta grafica”.

Una volta spiegate queste istruzioni, al soggetto veniva fatta apporre una firma nell'ultima colonna della tabella rappresentata in Fig. 4.1 (dove si attestava la presenza del paziente stesso) e un'altra firma su un foglio in cui dichiarava di aver appreso le istruzioni precedentemente spiegategli e di acconsentire all'esecuzione della prova.

Io sottoscritto.....MATEO MILANI.....
dichiaro di aver preso visione delle ISTRUZIONI
PER IL TEST e acconsento di prendervi parte.

Padova, 07 - 07 - 2010


Firma


Figura 4.2: Esempio di foglio di consenso, firmato da tutti i soggetti, per la partecipazione al test

SOGGETTO	TASK	ETA'	SESSO	BRACCIO	Distrazione sogg.	Manni alzate	IRREGOLARITA' NEI TEST		Interruzione task	FIRMA PRESENZA DEL SOGGETTO
							SI	Scatti pc		
1 NADIELE ROMANI	CE B F A D	27	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		X	12-30-31		Rob. Romani
2 ISMAELE DI GIULIO	EC B F A D	24	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		C	70		Sub. Ismaele
3 ALE PIAQUAN	DB C F A E	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		CA	10-31	X	ALE PIAQUAN
4 ENRICO CONTANZI	EC F B A D	22	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		AD	09-3		Enrico Contanzi
5 ANDREA CARLI	BC A F E D	26	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		C	2		Andrea Carli
6 MARCO FRISONI	CA E D B F	22	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		CAF	47-45		Marco Frisoni
7 LEA CAPELLA	FE D B C A	24	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		EDA	10-52-40		Lea Capella
8 VALENTINO DE FENER	D C E F B A	23	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		DFE	18-2-91		Valentino De Fener
9 PIRO SACCHETTI	A E B F D C	22	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		AE	4-32		Piro Sacchetti
10 LUCE NARDO	F D B E A C	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		D	10-30-40		Luce Nardo
11 MATTEO MILANI	B D F A C E	23	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		DFE	10-40-31		Matteo Milani
12 IDA TURELLI	C F E B D A	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)					Ida Turelli
13 CHIARA ZANATA	F E D A C B	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)	X	C	70-71		Chiara Zanata
14 PATRIZIA PANI	D F B A E C	21	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		B	52		Patrizia Pani
15 ANGELO ZAMPICINI	F A B D C E	29	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		C	50		Angelo Zampicini
16 MARINA CARRARO	C A F D E B	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		B	43		Marina Carraro
17 SILVIO CAPPIN	B D C F A E	22	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		E	50		Silvio Cappin
18 ALESSIO GARGIULO	E F C A D B	22	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)		F	31		Alessio Gargiulo
19 PAOLA NICOLI	E B C A F D	26	F	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)				X	Paola Nicoli
20 GIULIANA STANIA	D F A C E B	22	M	DX	INTEGRAZIONE TESTO IN UN DOCUMENTO (10-15)					Giuliana Stania

Figura 4.3: Tabella di Fig. 4.1 compilata e firmata da ciascun soggetto

4.2 Esecuzione Task

Mentre il soggetto svolgeva il test, si prestava molta attenzione al fatto che non fossero commesse irregolarità dal soggetto stesso o si verificassero dei problemi legati al funzionamento del computer.

Prima di iniziare questi test, infatti, ci eravamo accorti che nell'esecuzione di alcuni esercizi si verificano degli "scatti" nella visualizzazione da parte del computer. In questi casi il target faceva degli spostamenti molto "scattosi". Come conseguenza il soggetto che inseguiva il "pallino rosso" compiva degli "strappi" per cercare di raggiungere il target, aumentando così il proprio errore sia per quanto riguarda la distanza che la velocità rispetto al target stesso. Abbiamo cercato di risolvere questo problema diminuendo la frequenza del timer, passando da 100 Hz a 50 Hz. Così facendo il numero degli "scatti" del computer era notevolmente diminuito però in alcuni casi si verificava ancora. Proprio per questo motivo tali anomalie del sistema venivano segnalate sulla tabella rappresentata in Fig. 4.4 in modo che, al momento dell'analisi dei dati, si potesse tenere conto di questo fattore.

PIANIFICAZIONE TASKS - Generazione di feedback sonoro nell'esecuzione di task motori

	SOGGETTO	TASK	ETA'	SESSO	BRACCIO	IRREGOLARITA' NEI TEST					
						Distrazione sogg.	Mano alzata	Scatti pc		Interruzione task	
								Si	Istante scatto		Si
1	Romani Daniele	C E B F A D	24	M	dx	Ingresso persona nella stanza		B A D	68s 70s 37s		
2	Benedetti Giulio	E C B F A D	24	M	dx	Mano d'appoggio staccata (C-85s)		C	70s		
3	Pasqualin Alice	D B C F A E	22	F	dx	Perdita di controllo sul puntatore (A-115s)		C A	48s 37s	X	
4	Costanzi Enrico	E C F B A D	22	M	dx	Squillo del telefono (F-40s)		A D	105s 3s		
5	Carli Andrea	B C A F E D	26	M	dx	Ingresso persona nella stanza (C-15s)		C	2s		
6	Frison Marco	C A E D B F	22	M	dx			C D F	30s 47s 15s		
7	Caramma Lea	F E D B C A	21	F	dx	Cambio posizione mano d'appoggio (C-20s)		E D A	40s 52s 40s		
8	De Faveri Valentino	D C E F B A	23	M	dx			D F F	18s 2s 57s		
9	Sartorato Paolo	A E B F D C	22	M	dx			A E	4s 32s		
10	Nardo Alice	F D B E A C	22	F	dx	Il soggetto ha starnuto (D-15s)		D	40s		
08-lug											
11	Milani Matteo	B D F A C E	23	M	dx	Ingresso persona nella stanza (A-30s)		D E	40s 37s		
12	Maroni Ida	C F E B D A	22	F	dx						
13	Zanata Chiara	F E D A C B	22	F	dx		x (F)	C	70-75s		
14	Pavan Beatrice	D F B A E C	21	F	dx	Squillo del telefono (A-40s)		B	52s		
15	Zampicini Andrea	F A B D C E	29	M	dx	Ingresso persona nella stanza (B-75s)		C	50s		
16	Carraro Martina	C A F D E B	22	F	dx	Molto vicino al bordo sx della tavoletta (F)		D	43s		
17	Cardin Silvia	B D C F A E	22	F	dx			E	50s		
18	Garbujo Alessio	E F C A D B	22	M	dx			F	54		
19	Nicoli Paola	E B C A F D	26	F	dx					X	
20	Smania Giovanni	D F A C E B	22	M	dx						

Figura 4.4: Tabella finale con tutti i dati riguardo le anomalie durante i Task

A parte qualche piccola anomalia dovuta al sistema possiamo ritenerci soddisfatti di come si sono svolti i test. Tutti i soggetti sembravano aver affrontato i task con la giusta concentrazione e motivazione e le tempistiche sono sempre state rispettate.

Da come abbiamo potuto osservare mentre i soggetti svolgevano il test, soprattutto negli esercizi a traiettoria fissa (Task A, Task C, Task E), si presentava un effetto di learning. I soggetti, infatti, dopo pochi cicli iniziali, erano in grado di distinguere i due tipi di traiettoria e a prevedere quindi, in questo caso particolare, in quale punto il target avrebbe invertito il suo movimento. Questa condizione sembrava inoltre accentuarsi nel Task C, dove il feedback rispecchiava la velocità del target.

Dai commenti dei vari soggetti, appena terminato il test, è stato chiaro fin da subito che la modalità di audio che li ha aiutati maggiormente è stata quella legata alla velocità del target. L'altro tipo di feedback invece sembrava, ai più, una distrazione. I dati di ciascun soggetto sono stati salvati con la stringa *TaskLetteraTask_CognomeSoggetto* nella stessa cartella, pronti per essere analizzati.



Figura 4.5: Alcune fotografie dei test dell'otto luglio

Capitolo 5

Elaborazione dati test

5.1 Estrazioni dei dati significativi con Matlab

Premettendo fin da ora che l'analisi dei dati relativi ai test del 7-8 luglio sono ancora in corso. In questo capitolo cercherò di mettere in luce i primi dati interessanti estrapolati e trarrò delle conclusioni preliminari relativi a questi dati iniziali.

Prima di analizzare i dati con un software statistico abbiamo creato uno script in Matlab che permettesse di estrarre i valori interessati dalle struct in cui erano stati salvati.

Per questa prima analisi abbiamo deciso di confrontare tra loro i dati riguardanti le posizioni dei soggetti e del target di ciascun Task relativo a tutti i soggetti presi in esame. Su questi dati abbiamo calcolato due tipi diversi di errori:

- Errore relativo alla distanza media tra target e soggetto

$$d_m = \frac{1}{N} \sum_{i=1}^N d_i = \frac{1}{N} \sum_{i=1}^N \frac{\sqrt{(x_{T_i} - x_i)^2 + (y_{T_i} - y_i)^2}}{r_T}$$

Figura 5.1: Formula della distanza media tra target e soggetto

- Errore medio pesato con segno relativo alla posizione del target rispetto a quella del soggetto.

$$e_{m=} = \frac{1}{N} \sum_{i=1}^N e_i = \frac{1}{N} \sum_{i=1}^N \text{sign}(v_{T_i}) \frac{x_{T_i} - x_i}{r_T}$$

Figura 5.2: Formula dell'errore medio pesato con segno

Questi due tipi di errori, inoltre, sono stati normalizzati al valore del raggio del target (nel nostro caso pari a 25 pixel) e sempre per ciascuno di essi abbiamo calcolato la deviazione standard.

Per rendere il tutto più efficiente abbiamo fatto in modo che il nostro script stampasse i valori degli errori desiderati in un file .xls con un nome che rappresentasse il valore dell'errore che si sta considerando e alcuni parametri (come l'inizio e la fine del ciclo preso in esame). Si ottiene, ad esempio, un file rappresentato dalla stringa: 1_28_risprova_distanza.xls.

L'errore relativo alla distanza media tra target e soggetto ci fornisce un'indicazione di quanto il paziente si sia allontanato dal "pallino" del target durante l'esecuzione del Task. Per trovare questo parametro ci siamo serviti della funzione `pdist`, presente nella libreria di Matlab, che permette di calcolare la distanza tra una coppia di oggetti.

Da un'analisi preliminare dei grafici (riportati in Figura 5.3) rappresentanti l'errore sulla distanza media, si evince che la traiettoria a lunghezza fissa ha presentato molte meno difficoltà nell'esecuzione rispetto a quella a lunghezza variabile, in tutte le modalità di audio a disposizione.

Si potrebbero trarre delle prime conclusioni anche per quanto riguarda un confronto tra modalità di esercizio con la stessa traiettoria ma con audio diverso. Tuttavia tali informazioni non sarebbero significative da un punto di vista qualitativo e quantitativo.

Per questo tipo di conclusioni quindi utilizzeremo un software statistico che ci potrà dare informazioni di sicuro più precise a riguardo.

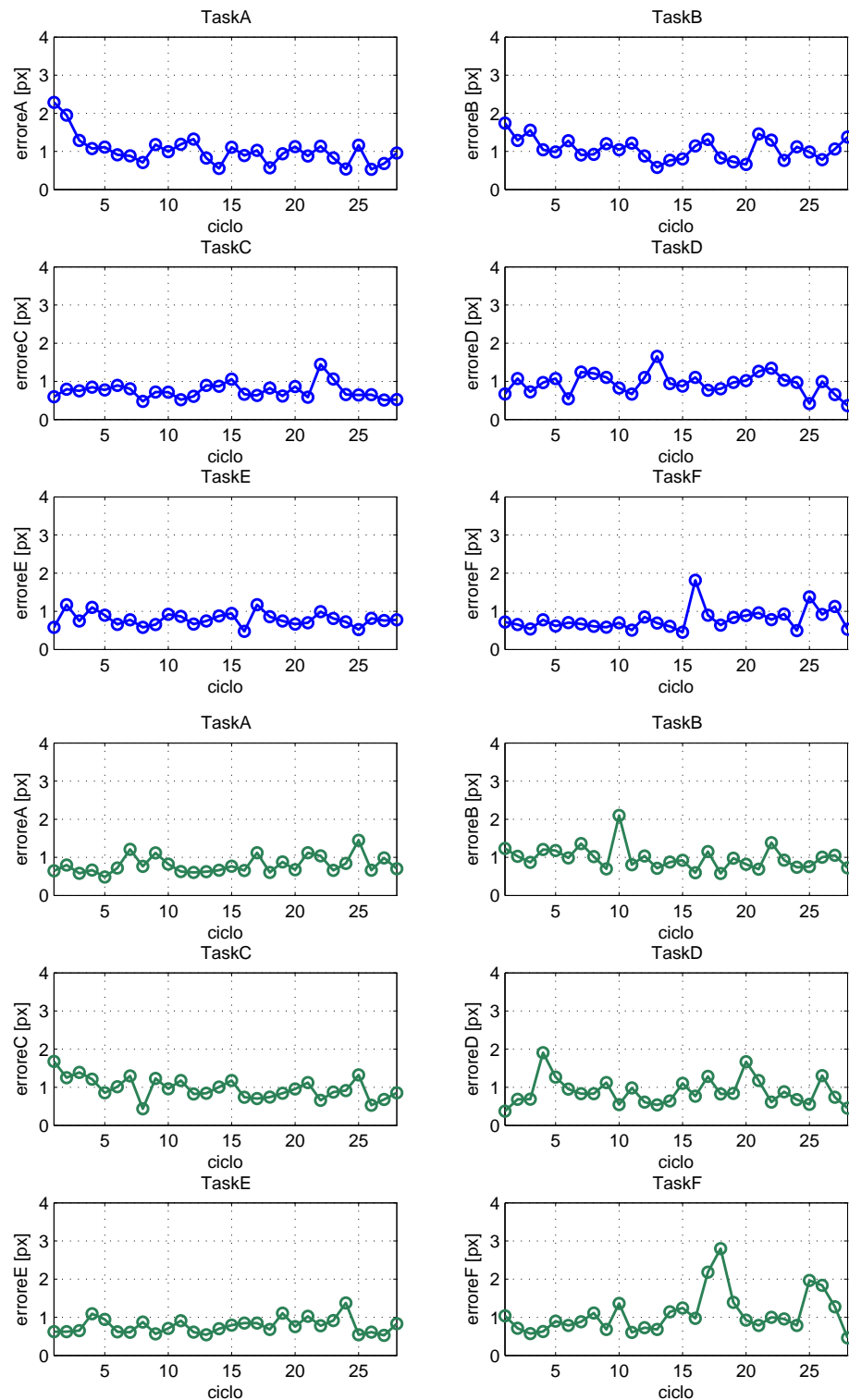


Figura 5.3: Grafici Matlab di 2 dei soggetti relativi all'errore legato alla distanza tra target e soggetto

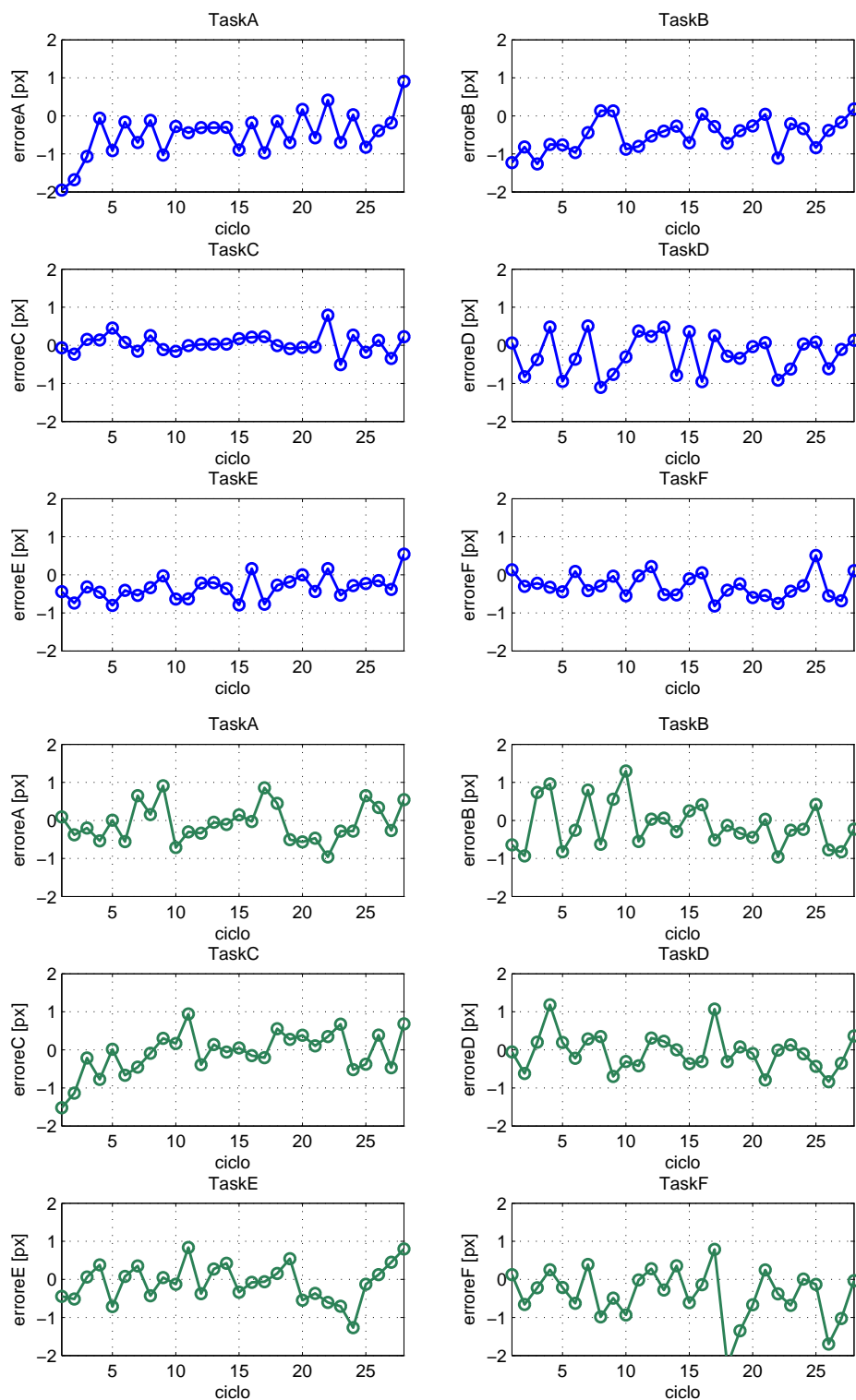


Figura 5.4: Grafici Matlab di 2 dei soggetti relativi all'errore medio pesato con segno

Per quanto riguarda invece l'errore medio pesato con segno, esso è stato molto utile per capire quanto un soggetto fosse in ritardo o in anticipo rispetto al target. I soggetti, infatti, seguendo il "pallino rosso" tendevano principalmente ad anticipare i movimenti di questo, restando quindi sempre in ritardo rispetto al movimento del target stesso.

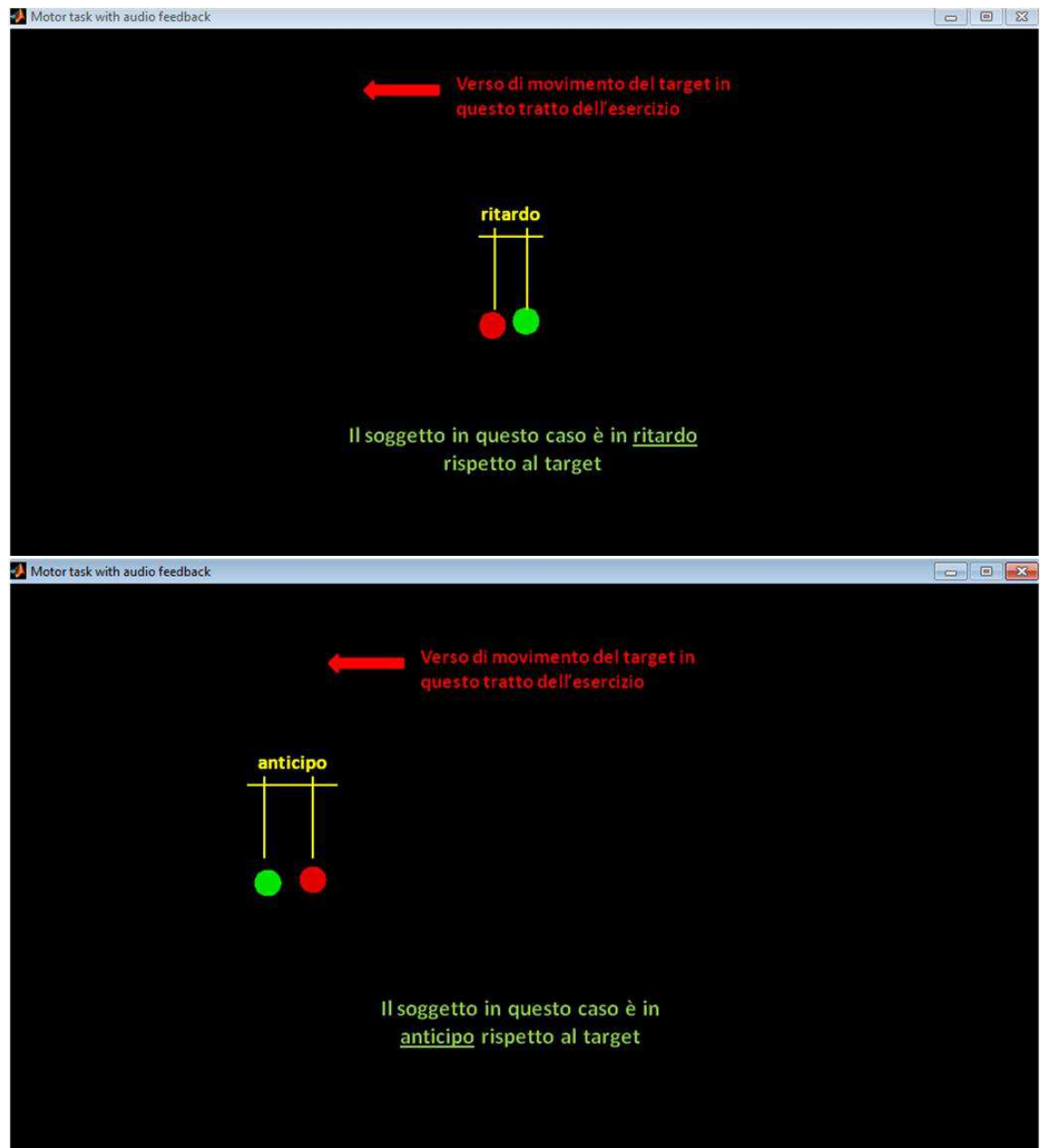


Figura 5.5: Illustrazione caso ritardo e anticipo del soggetto rispetto al target

Come si può notare dai grafici in Fig. 5.4, anche l'errore medio pesato con

segno mette in evidenza che i tracciati dei Task a lunghezza fissa sono migliori rispetto a quelli che raffigurano una traiettoria a lunghezza variabile.

Un'altro fatto molto interessante da notare in questi grafici è che si può vedere un miglioramento anche tra esercizi con la stessa traiettoria ma con modalità audio diversa. Un esempio lampante è la differenza tra i Task A e C del soggetto a cui appartengono il primo gruppo di grafici di Fig. 5.4. Ciò significa che in questo particolare caso il feedback audio ha agevolato il soggetto nell'esecuzione del Task.

5.2 Elaborazione dei dati con software statistico

5.2.1 GraphPad Prism

GraphPad Prism è un software statistico che combina grafica scientifica, curve di interpolazione e statistiche in un programma molto semplice da utilizzare. Prism ha tutte le capacità che ci si aspetterebbe da un programma scientifico di prima qualità grafica, ma ciò che lo rende veramente unico è il fatto che fornisce un'interfaccia utente-software veramente molto intuitiva e ricca di possibilità.

Tramite questo programma si possono creare diverse funzioni statistiche in grado di aggiornarsi istantaneamente nel momento in cui si va a cambiare i dati dalla tabella di origine. Lo stesso concetto vale anche per la creazione dei grafici.

5.2.2 Test di Wilcoxon

Il test di Wilcoxon-Mann-Whitney noto pure come test U di Mann-Whitney è uno dei più efficaci test non parametrici (test di verifica di ipotesi).

Viene utilizzato per verificare, in presenza di valori ordinali appartenenti ad una distribuzione continua (ovvero che possiedono una relazione di ordine totale), se due campioni statistici provengono dalla stessa popolazione.

La potenza-efficienza di questo test è attorno al 95% sia per campioni piccoli che per campioni grandi.

Questo test permette, quindi, di ottenere maggiori osservazioni in una popolazione rispetto ad un'altra. L'ipotesi nulla (affermazione sulla distribuzione di

probabilità di una o più variabili casuali) nel test di Mann–Whitney è quella che i due campioni siano tratti da una popolazione singola. Per questa ragione le loro distribuzioni di probabilità sono uguali.

L'ipotesi alternativa è che uno dei campioni sia più grande in maniera stocastica. Questo richiede che i due campioni siano statisticamente indipendenti e l'osservazione sia ordinale (ad esempio si può almeno dire di qualsiasi due osservazioni, quale sia la maggiore).

Nel nostro caso particolare abbiamo comparato diversi Task (ritenuti da noi più significativi) tra loro, con il criterio di Wilcoxon, per vedere se si ottenevano dei valori significativi dalle loro differenze.

5.2.3 Il nostro caso

I dati riguardo l'errore sulla distanza media con relativa deviazione standard e l'errore medio pesato con segno, sempre con la relativa deviazione standard, precedentemente calcolati in Matlab (vedi paragrafo 5.1), sono stati inseriti in GraphPad Prism per una rielaborazione più approfondita. Per ciascun tipo di errore abbiamo creato una tabella che mette in risalto tutti i nomi dei soggetti (ordinati in ordine alfabetico) ciascuno correlato ai propri errori e deviazioni standard in ogni Task previsto dal test (Fig. 5.6).

Table format: Grouped	A		B		C		D		E		F		
	Task A		Task B		Task C		Task D		Task E		Task F		
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
1	BENEDETTI GIULIO	0.883581	0.508671	1.091563	0.725283	0.916558	0.533270	1.011943	0.654802	1.119704	0.724487	1.088848	0.768275
2	CARAMMA LEA	1.028777	0.637357	1.240739	0.993692	1.123007	0.779648	0.998302	0.692021	1.208299	0.793938	1.387259	1.052197
3	CARDIN SILVIA	1.051206	0.660057	1.686266	1.173140	1.240092	0.675778	1.412016	1.012854	1.266252	0.653065	1.528968	1.065877
4	CARLI ANDREA	1.263232	0.866603	1.806604	1.669939	1.076044	0.747928	1.564437	1.148716	1.060079	0.683786	1.280561	0.803180
5	CARRARO MARTINA*	1.724549*	1.604082*	1.212178*	0.799320*	2.228731*	0.913078*	1.226668*	0.690566*	1.169831*	0.663774*	1.825395*	1.225645*
6	COSTANZI ENRICO	0.902098	0.582994	0.911766	0.687269	0.822447	0.512859	0.925529	0.652321	0.874578	0.596460	0.977973	0.664371
7	DE FAVERI VALENTINO	1.041902	0.703636	1.183481	0.827712	0.965127	0.638942	1.238423	0.813635	1.047460	0.676126	1.153692	0.774112
8	FRISON MARCO	1.036939	0.672278	0.975468	0.628431	1.069805	0.747059	0.979155	0.723050	1.002743	0.694222	1.016015	0.719340
9	GARBUJO ALESSIO	1.046620	0.680714	1.047585	0.616418	1.141456	0.678494	1.093642	0.647039	1.574494	0.900960	1.427806	0.968405
10	MARONI IDA	0.802585	0.542398	0.961085	0.695697	0.975197	0.606335	0.862945	0.652805	0.774997	0.497469	0.970132	0.680872
11	MILANI MATTEO	0.978998	0.732033	1.158631	0.800706	0.878770	0.653597	1.039013	0.675945	0.883669	0.592078	1.256656	0.840787
12	NARDO ALICE	1.009510	0.647371	1.203675	0.696408	0.929492	0.610471	1.449776	0.929384	1.082960	0.669722	1.295006	0.774156
13	NICOLI PAOLA	1.057881	0.732706	1.192325	0.807040	1.057077	0.797225	1.193484	0.799947	1.060462	0.791331	1.299701	0.853696
14	PASQUALIN ALICE*	0.000000*		1.122468*	0.805582*	0.925376*	0.610786*	1.323214*	0.926342*	0.878437*	0.546931*	1.055917*	0.727699*
15	PAVAN BEATRICE	1.121347	0.772186	1.072703	0.735195	0.876372	0.531659	1.818390	1.217252	1.127681	0.819255	1.171406	0.829627
16	ROMANI DANIELE	0.844431	0.592128	1.055504	0.823892	0.803190	0.541187	0.920319	0.610306	0.846615	0.516213	0.953685	0.636195
17	SARTORATO PAOLO	1.022625	0.724304	1.024490	0.728312	0.749074	0.477973	0.887566	0.600142	0.793023	0.498667	0.739634	0.464803
18	SMANIA GIOVANNI	1.150185	0.748841	1.189763	0.862043	1.036839	0.664774	1.325351	0.916865	0.918359	0.636135	1.319636	0.820198
19	ZAMPICININI ANDREA	0.844693	0.550691	1.018869	0.706468	0.911072	0.624157	0.999707	0.659882	0.868758	0.526555	1.115768	0.727351
20	ZANATA CHIARA	1.418375	0.871530	1.289556	0.883132	1.302391	0.777696	1.273698	0.720093	1.100892	0.592303	1.646327	1.042798

Figura 5.6: Tabella ottenuta con GraphPad Prism

Una volta implementata la tabella, ci siamo chiesti quali potessero essere i test più significativi che potevano essere messi a confronto.

Dato che lo scopo del mio elaborato era quello di valutare se un feedback audio potesse agevolare l'esecuzione dell'esercizio o meno, sicuramente un confronto utile sarebbe stato quello tra i Task: A-C, A-E, B-D, B-F. Un altro confronto molto interessante è stato quello tra Task con la stessa modalità di audio, ma con traiettoria differente, tipo: A-B, C-D, E-F.

Sfruttando il criterio di Wilcoxon abbiamo provveduto quindi a elaborare queste differenze. Da una prima analisi ci siamo subito accorti che i valori ottenuti erano troppo distanti dai valori da noi ipotizzati. Siamo andati quindi a rianalizzare i dati relativi agli errori di ogni singolo soggetto per verificare se ci fossero delle anomalie. Dopo un'accurata analisi abbiamo concluso che 2 dei 20 soggetti presi in esame (Carraro Martina e Pasqualin Alice) dovevano essere considerati outlier perché presentavano dei valori troppo distanti dalla media generale degli altri elementi e quindi aberranti nelle medie conclusive. Voglio sottolineare che questi 2 soggetti sono stati esclusi nei tre grafici e tabelle qui di seguito.

Fatta questa selezione abbiamo ripetuto il test di Wilcoxon e abbiamo ottenuto dei primi risultati molto interessanti.

CICLI CONSIDERATI PER CIASCUN TASK: DA 1 A 28

	ERRORE DISTANZA MEDIA			ERRORE MEDIO PESATO CON SEGNO		
	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []
A-C	NO			SI	0,002	0,035054128
A-E	NO			NO		
B-D	NO			NO		
B-F	NO			NO		
A-B	SI	0,0035	-0,144393711	NO		
C-D	SI	0,0069	-0,173315861	NO		
E-F	SI	0,002	-0,167669306	NO		

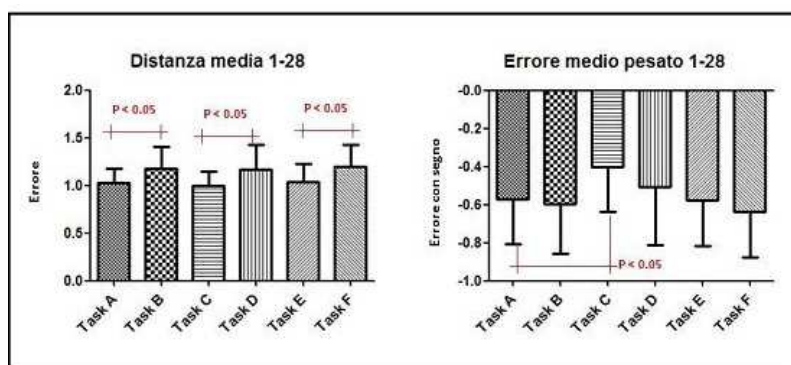


Figura 5.7: Tabella e grafici di analisi dati per i 28 cicli di ciascun Task

Come si può vedere dalla tabella in Fig. 5.7 e dai grafici (sempre in Fig. 5.7), i due diversi tipi di errori hanno messo in evidenza due differenze nell'esecuzione dei vari task.

Il primo risultato da segnalare è che, in tutti i tipi di test, l'errore nei Task con traiettoria a lunghezza fissa è sempre minore rispetto a quelli caratterizzati da traiettoria a lunghezza variabile. Ciò può dipendere dal fatto che l'imprevedibilità del moto del target negli esercizi a lunghezza variabile metteva in seria difficoltà i soggetti nel seguirlo. Questo è dimostrato anche dal fatto che la differenza tra i Task A-B e i Task C-D risulta significativa.

Analizzando ora i Task in funzione del feedback audio, si può vedere come nel grafico "Errore medio pesato 1-28" l'audio legato alla velocità del target abbia ridotto l'errore dal Task A al Task C (in questo caso infatti A-C risulta significativa).

Nonostante questi primi risultati fossero già abbastanza soddisfacenti, ci siamo chiesti per quale motivo non risultasse significativa anche la differenza B-D nel grafico "Errore medio pesato 1-28". Ci aspettavamo, infatti, che la modalità di audio legata alla velocità del target dovesse produrre degli esiti significativi anche nelle traiettorie a lunghezza variabile.

Abbiamo quindi deciso di analizzare dei nuovi dati (sulla base anche dei grafici Matlab in Fig. 5.3) , escludendo per ciascuno dei soggetti i primi 10 cicli del Task con cui ogni soggetto ha iniziato il test. Ricordo che, per il processo di randomizzazione, ciascun soggetto partiva con un Task diverso.

Questa scelta è stata dettata dal fatto che secondo noi i primi 10 cicli del primo Task per ciascun paziente sono serviti più che altro ad apprendere come funziona il test in se stesso.

CICLI CONSIDERATI PER CIASCUN TASK: DA 1 A 28 esclusi 10 cicli iniziali del Task di partenza

	ERRORE DISTANZA MEDIA			ERRORE MEDIO PESATO CON SEGNO		
	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []
A-C	NO			SI	0,0011	0,036412956
A-E	NO			NO		
B-D	NO			NO		
B-F	NO			NO		
A-B	SI	0,0027	-0,161792959	NO		
C-D	SI	0,0041	-0,152644322	NO		
E-F	SI	0,0004	-0,1821785	NO		

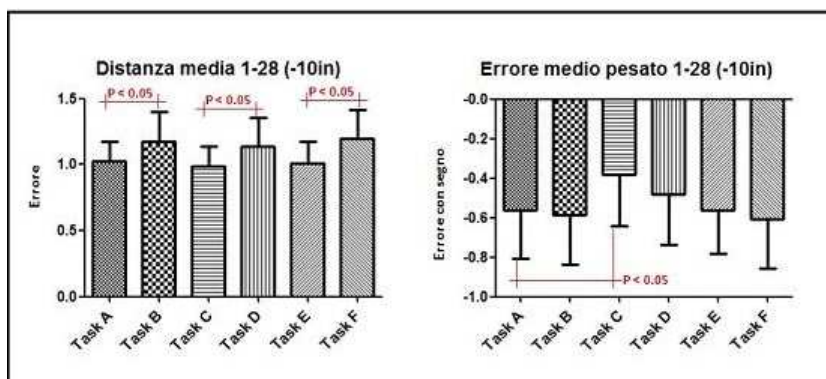


Figura 5.8: Tabella e grafici di analisi dati per i 28 cicli di ciascun Task esclusi i primi 10 cicli del Task iniziale di ogni soggetto

I grafici in questo caso rispecchiano quelli precedenti e nonostante ci sia un miglioramento nei valori di P (valori più piccoli dei precedenti) non siamo riusciti a mettere in evidenza che l'audio legato alle velocità del target abbia portato qualche beneficio ai soggetti nel passaggio dal Task B al Task D. Per questo motivo abbiamo continuato la nostra ricerca e abbiamo deciso di escludere interamente dall'analisi il Task con cui ciascun soggetto ha iniziato il proprio test. In questo modo siamo riusciti finalmente ad ottenere dei risultati significativi per quanto riguarda il feedback audio.

Come si può vedere quindi da questi ultimi risultati (in Fig.5.9), anche la differenza B-D è diventata significativa (come si può vedere nel grafico "Errore medio pesato wf"). Ciò significa che, con questa particolare scelta di escludere il primo Task dall'analisi dei dati, siamo riusciti a dimostrare che il feedback audio legato alla velocità del target applicato alla traiettoria a lunghezza variabile ha prodotto un risultato significativo nel ridurre l'errore.

CICLI CONSIDERATI PER CIASCUN TASK: DA 1 A 28 escluso il primo Task di ciascun soggetto

	ERRORE DISTANZA MEDIA			ERRORE MEDIO PESATO CON SEGNO		
	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []	SIGNIFICATIVO	Valore di P	Valore differenza media $\Delta\mu$ []
A-C	NO			SI	0,004	0,036412956
A-E	NO			NO		
B-D	NO			SI	0,0269	0,034713133
B-F	NO			NO		
A-B	SI	0,0166	-0,0889398	NO		
C-D	NO			NO		
E-F	SI	0,0024	-0,16695064	NO		

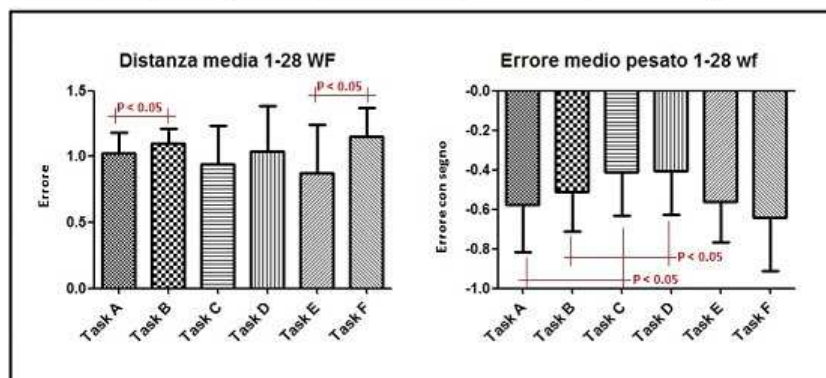


Figura 5.9: Tabella e grafici di analisi dati per i 28 cicli di ciascun Task escluso il primo Task eseguito da ogni soggetto

Conclusioni

Al termine di questo mio lavoro, posso affermare che le consegne iniziali e i tempi previsti per la realizzazione dell'elaborato sono stati pienamente rispettati.

La parte più impegnativa è stata sicuramente quella di realizzare l'interfaccia GUI in Matlab che rispondesse a tutti i requisiti richiesti e non presentasse anomalie durante l'esecuzione dei test.

Per dei progetti futuri la nostra GUI potrà senza dubbio essere migliorata aggiungendo qualche traiettoria nuova, oltre alle due già esistenti. Si potranno implementare, inoltre, nuove tipologie di feedback audio che siano in grado di dare un maggior contributo al soggetto nel raggiungimento del target.

Per quanto riguarda i risultati ottenuti finora, come già preannunciato all'inizio del capitolo 5, sono da considerarsi solo come esiti preliminari.

Rielaborando nuovamente i dati, con nuovi metodi e parametri (intervalli di cicli più stretti e.g. 5-22 e diverse tipologie di errore) si noteranno certamente analogie significative tra i task presi in esame.

Gli esiti comunque finora ottenuti, e qui illustrati nel capitolo 5, fanno ben sperare per quanto riguarda l'utilizzo di un feedback audio in un esercizio riabilitativo. Si è visto infatti che il feedback audio legato alla velocità del target permetteva ai soggetti di ottenere dei significativi miglioramenti rispetto all'esecuzione del medesimo esercizio però privo di audio.

Sarebbe dunque molto interessante testare la nostra interfaccia con il feedback di cui ho appena parlato su pazienti post-ictus allo scopo di vedere se effettivamente questo strumento può agevolare il paziente a svolgere e terminare l'esercizio riabilitativo con migliori risultati per il recupero funzionale dell'arto lesso.

Bibliografia

- [1] T. I. L. S. W. Group, “Prevalence of chronic diseases in older italians: comparing self reported and clinical diagnoses,” *Australian Journal of Physiotherapy*, vol. 1002, pp. 26–995, 1997.
- [2] A. Caperna, “Ictus, percorsi assistenziali contro la malattia, la proposta del ministro fazio,” *Salute Domani*, 2010.
- [3] C. A, M. C, and D. N. M, “High stroke incidence in the prospective community-based,” *Aquila registry 1994-1998*, vol. Stroke 1997, no. 6, pp. 28–2500, 1997.
- [4] D. Carlo, A. Baldereschi, M. G. C, and al, “Stroke in elderly population: incidence and impact on survival and daily function,” *The Italian Longitudinal Study on Aging*, vol. Cerebrovascular Disease, no. 50, pp. 16–141, 2003.
- [5] R. Colombo, F. Pisano, S. Micera, A. Mazzone, C. Delconte, M. C. Carrozza, P. Dario, and G. Minuccio, “Robotic techniques for upper limb evaluation and rehabilitation of stroke patients,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 13, no. 3, pp. 313–324, 2005.
- [6] E. Carraro, S. Masiero, G. Rosati, and C. Ferraro, “Recupero della forza muscolare dopo precoce mobilizzazione passiva nel paziente emiplegico,” *EUROPA MEDICOPHYSICA*, vol. 44, no. 3, pp. 1–3, 2008.
- [7] A. Roda, F. Avanzini, S. Masiero, , and G. Rosati, “Auditory feedback in robot-assisted neuro-rehabilitation: state of the art and future prospects,” -, vol. -, no. -, pp. -, 2010.

-
- [8] M. H. Thaut, G. C. MacIntosh, and R. R. Rice, "Rhythmic facilitation of gait training in hemiparetic stroke rehabilitation," *Clin rehabil*, vol. 17, no. -, pp. 713–722, 2003.
- [9] S. Koelsch, E. Kasper, D. Sammler, K. Schulze, T. G. T, and A. D. Friederici, "Music, language and meaning: brain signatures of semantic processing," *Nat Neurosci*, vol. 7, pp. 302–307, 2004.
- [10] K. E. Watkins, A. P. Strafella, and T. Paus, "Seeing and hearing speech excites the motor system involved in speech production," *Neuropsychologia*, vol. 41, no. 9, pp. 989–994, 2003.
- [11] R. A. Magill, "Motor learning and control: Concepts and applications," *New York: McGraw-Hill*, no. 7th Edition, 2003.
- [12] P. M. V. Vliet and G. Wulf, "Extrinsic feedback for motor learning after stroke: What is the evidence," *Disability and Rehabilitation*, vol. 28, no. -, pp. 831–840, 2006.
- [13] R. Secoli, G. Rosati, and J. Reinkensmeyer, "Using sound feedback to counteract visual distractor during robot-assisted movement training," -, vol. -, no. -, pp. -, 2009.